ISSN: 0393 - 1439

CHIP SPECIAL

Mensile di
micro
e personal
computer
N° 10 Novembre 1985
L. 7000

Commodore 64 PROGRAMMI

4

casa editrice tecniche nuove - spedizione in abbonamento postale-gruppo III/70

Giochi
Grafica, Musica
Organizzazione, Hobby

tecniche nuove

Introduzione

Questa edizione speciale risponde al Vostro desiderio di tanti e soprattutto di buoni programmi applicativi, che servano a facilitare la scrittura nel linguaggio più veloce. In questo numero infatti troverete un tool-set indispensabile per l'attività di programmazione: un bel programma Monitor e un Assembler a macro particolarmente valido!
Per quelli di Voi che finora non hanno mai affrontato l'Assembler e i listati misti, in BASIC e in codice macchina, ecco alcuni preziosi consigli:

— Innanzitutto introducete il listato BASIC e memorizzatelo su disco o cassetta.
— Poi introducete la parte di caricamento del codice macchina. Per evitare gli eventuali errori di input, i programmatori hanno sviluppato delle speciali routine; se le utilizzate sarà lo stesso computer a segnalarvi gli errori.
— L'indirizzo iniziale e finale dei programmi in codice macchina (MC = machine code) si trovano nella prima linea del listato. Anche questo programma MC va memorizzato su un supporto dati (LOAD "Nome del programma" INDIRIZZO INIZIALE, LUNGHEZZA DEL PROGRAMMA).
— Ricordatevi di memorizzare completamente i vostri programmi prima di lanciarli, cioè prima di dare il RUN, altrimenti (in caso di errore) tutto il lavoro di input rischia di andare perso.

Per i programmi particolarmente importanti per tutti gli utenti del C64, abbiamo messo a disposizione il "Dottore dei dischetti", che permette di salvare i programmi che per un motivo o per l'altro si sono rovinati.
A questi programmi applicativi sono affiancati programmi di divertimento, giochi di avventura, concettuali e da casinò.

La redazione di CHIP Special

CHIP SPECIAL

mensile di programmi per micro e personal computer

Direttore responsabile: Giuseppe Nardella. Rivista mensile registrata il 16 giugno 1984 al numero 296 presso il Tribunale di Milano. Registro nazionale della stampa n. 630 v. 7 f. 233 - 16.12.82. **Distributore:** Messaggerie Periodici ME.PE., Via G. Carcano, 32 - 20141 Milano. **Stampa:** Arti Grafiche F.lli Jodice, Rosate (MI). **Fotocomposizione:** Grafica Quadrifoglio, Milano.
© per la Germania: Vogel Verlag, Würzburg; © per l'Italia: Tecniche Nuove, via Moscova 46/9A, 20121 Milano, tel. (02) 6590351, telex 334647 TECHS I. Spedizione in abbonamento postale gruppo III/70.

Indice

Notizie utili
Simboli grafici sul video 3

Giochi concettuali
Esagono 4

Giochi di carte
Casinò-Blackjack 15

Giochi di avventura
La terra dei 7 anelli 20

Linguistica
Le coniugazioni latine 30

Grafica
Istogrammi 35

Pubblicità
Scritte scorrevoli 38

Trucchi e consigli
Caricatore HEX 52
Rat-Tape 54
Screensave 57

Strumento di salvataggio
Dottore dei dischetti 60

Tool-Set
CHIP Monitor 69
Macro Assembler/Editor 82

Simboli grafici sul video

Nel vostro manuale non sono trattati i simboli grafici.

Quando sul video appare il seguente simbolo		Significato
	SHIFT CLR/HOME	cancellare il video
	CLR/HOME	cursore all'inizio del video
	CTRL 0	reverse disattivato
	CTRL 0	reverse attivato
	SHIFT ↑ CRSR ↓	cursore in alto
	↑ CRSR ↓	cursore in basso
	SHIFT ← CRSR →	cursore a sinistra
	← CRSR →	cursore a destra
	CTRL 1	colore nero
	CTRL 2	colore bianco
	CTRL 3	colore rosso
	CTRL 4	colore turchese
	CTRL 5	colore violetto
	CTRL 6	colore verde
	CTRL 7	colore blu
	CTRL 8	colore giallo

Quando sul video appare il seguente simbolo		Significato
	CTRL 1	colore arancio
	CTRL 2	colore marrone
	CTRL 3	colore rosso chiaro
	CTRL 4	colore grigio 1
	CTRL 5	colore grigio 2
	CTRL 6	colore verde chiaro
	CTRL 7	colore blu chiaro
	CTRL 8	colore grigio 3
		liberamente definibile
		liberamente definibile
		liberamente definibile
		liberamente definibile
		liberamente definibile
		liberamente definibile
		liberamente definibile
		liberamente definibile

Esagono

Lunghezza del programma: 11000 byte. Linguaggio di programmazione: BASIC e Assembler. Accessori: nessuno. Giocatori: 4.

"Scrivendo questo programma mi sono soprattutto preoccupato di garantire una partecipazione attiva. Inoltre volevo creare un gioco che non soltanto fosse adatto ai "computer freak", ma che rappresentasse un divertimento per un gruppo di amici", così ci ha scritto l'autore di questo programma. Si gioca in quattro, in due gruppi di due giocatori; un altro giocatore funge da giudice di gara.

LE REGOLE DEL GIOCO

Un giocatore di un gruppo siede davanti al monitor, mentre il suo compagno è sistemato in modo da non poter vedere il video. Il giocatore davanti allo schermo deve far sì che il suo compagno riesca ad indovinare delle parole.
Si sceglie uno di sei esagoni e sullo schermo appaiono sei "argomenti", tra cui se ne deve selezionare uno. Sullo schermo appare il primo "vocabolo", cioè la parola da indovinare. Il giocatore davanti al video deve descriverla al suo compagno, in modo che questi possa capire di che cosa si tratta. Quando si arriva alla risposta esatta, il giudice preme il tasto "S" e si prosegue.
Una volta indovinato il secondo, il terzo ..., il sesto vocabolo, il primo gruppo di giocatori ha finito il suo primo giro. Ricompaiono gli argomenti, meno quello scelto in precedenza; i due giocatori si scambiano i ruoli, cioè il giocatore che prima doveva indovinare, adesso siede davanti al video e si passa al secondo gruppo di vocaboli.
Quando sono stati passati in rassegna tutti e sei gli argomenti, finisce il turno del primo gruppo di giocatori e inizia a giocare il secondo gruppo.

COSA SUCCEDE

Scegliendo per esempio l'esagono numero 2, compaiono i seguenti argomenti: 1. automobili, 2. animali esotici, 3. lago/palude, 4. bevande, 5. verdure, 6. vacanze.
Il gruppo numero 1 seleziona per esempio 1; il video si pulisce ed appare il primo vocabolo relativo alla voce "automobili": tettuccio apribile.
Il giocatore che siede davanti al video deve descrivere questa parola al suo compagno; potrebbe dire per esempio:

"Un'apertura per il passaggio di aria..."

Non appena il compagno individua la parola e la dice, il giudice preme il tasto "S", al gruppo viene assegnato un punto e appare la seconda parola da indovinare.

NON BARATE

Non è permesso nominare neanche in parte la parola da indovinare (non si può dire per esempio "un tetto che..."). In questo caso il giudice preme il tasto "N" e si passa alla parola successiva senza totalizzare il punto. Inoltre bisogna ricordare che il tempo a disposizione per ogni giro è limitato; se per esempio il tempo scade alla quarta parola indovinata, non si possono più totalizzare altri punti indovinando anche le parole cinque e sei. Un ultimo consiglio: se il giocatore davanti al video vede che il suo compagno è in difficoltà, può dire "avanti". Il giudice preme il tasto "N" e si passa alla parola successiva; in questo modo avete la possibilità di indovinarne un'altra prima che scada il tempo. Anche l'altro giocatore può dire "avanti", se si accorge di essere fuori strada.

IL PROGRAMMA

Naturalmente giocare sempre con le stesse parole non è molto divertente. L'utente del programma quindi, quello che fa da giudice e conosce tutte le parole, ha la possibilità di aggiungere nuovi argomenti e vocaboli.

Questa aggiunta avviene alle linee DATA da 1190 a 1690.
Bisogna fare attenzione che le linee DATA siano in gruppi di sette, cioè la linea uno per gli argomenti e le linee da due a sette per i rispettivi vocaboli.
Le linee DATA non devono contenere necessariemente dei vocaboli, ma anche modi di dire o proverbi. La vostra fantasia non deve conoscere limiti, cercate solo di trovare un giusto equilibrio tra esagoni "troppo semplici" ed esagoni "estremamente difficili".
Infine avete anche la possibilità di cambiare il tempo a disposizione per ciascun giro; basta cambiare alla linea 2730 il valore per cui viene diviso TI (attualmente: 60). Il valore di 60 della versione originale corrisponde ad un tempo di 30 secondi circa; aumentando il valore, il tempo si allunga; diminuendolo, il tempo si accorcia.

Bodo Geschwindner

```
1 REM ESAGONO (PRE PROGRAMMA) PER CHIP SPECIAL
2 REM DI B. GESCHWINDNER QUESTO PROGRAMMA DEVE ESSERE
3 REM CARICATO E LANCIATO PRIMA DEL PROGRAMMA PRINCIPALE.
4 REM SI CANCELLA AUTOMATICAMENTE
5 REM ****************************
10 POKE44,11:POKE11*256,0
20 PRINTCHR$(147):POKE53280,0:POKE53281,0
22 -.
30 PRINT"■■■■■■ORA CARICATE E LANCIATE IL PROGRAMMA PRINCIPALE"
35 PRINT"■LOAD 'ESAGONO'   0    LOAD   'ESAGONO',8"
40 PRINT"■+L (TURBO) 0    LOAD (NORMALE) 'ESAGONO'"
50 PRINT,"■SECONDO LA VERSIONE!"
60 NEW


70 REM ESAGONO (PROGRAMMA PRINCIPALE)
80 REM PER CHIP SPECIAL
90 REM DI B. GESCHWINDNER
100 POKE53280,0:POKE53281,0:PRINTCHR$(147)
110 PRINTCHR$(158)TAB(15)"_________"
120 PRINTTAB(14)"/"TAB(23)"\"
130 PRINTTAB(13)"/"TAB(24)"\"
140 PRINTTAB(12)"/"TAB(25)"\"
150 PRINTTAB(11)"/"TAB(26)"\"
160 PRINTTAB(10)"/"TAB(27)"\"
170 PRINTTAB(9)"/"TAB(28)"\"
180 PRINTTAB(8)"/"TAB(29)"\"
190 PRINTTAB(7)"/"TAB(30)"\"
200 PRINTTAB(6)"/"TAB(31)"\"
210 PRINTTAB(5)"/"TAB(32)"\"
220 PRINTTAB(4)"/"TAB(33)"\"
230 PRINTTAB(4)"\"TAB(33)"/"
240 PRINTTAB(5)"\"TAB(32)"/"
250 PRINTTAB(6)"\"TAB(31)"/"
260 PRINTTAB(7)"\"TAB(30)"/"
270 PRINTTAB(8)"\"TAB(29)"/"
280 PRINTTAB(9)"\"TAB(28)"/"
290 PRINTCHR$(154)TAB(4)"ASPETTATE ■_____________"
300 :
310 :
320 REM    (C) TECNICHE NUOVE
330 REM        MOSCOVA 46/9A
340 REM        20121 MILANO
350 REM    TEL:(02) 65.90.351
360 REM
```

```
370 :
380 :
390 GOSUB3090:GOTO 3580
400 PRINTCHR$(147):GOSUB5490
410 EN=0:ZU=1:REM DRIVE NUM. 1/2
420 IFRND(X)>.5THENZU=3
430 :
440 REM....VIDEATA + NOMI
450 :
460 PRINTCHR$(147):PRINTCHR$(5)
470 PRINT"■IN QUESTO GIOCO"
480 PRINT"■■■■■■GAREGGIONO DUE GRUPPI OPPOSTI."
490 PRINT"■IL GRUPPO CHE TOTALIZZA IL MAGGIOR NUMERO DI PUNTI, VINCE!"
500 PRINT"■I NOMI DEI QUATTRO GIOCATORI SONO :"
510 PRINTCHR$(158)
520 INPUT"    GRUPPO 1/1.NOME : ";N$(1)
530 GOSUB3390
540 INPUT"    GRUPPO 1/2.NOME : ";N$(2)
550 GOSUB3390
560 PRINTCHR$(30)
570 INPUT"    GRUPPO 2/1.NOME : ";N$(3)
580 GOSUB3390
590 INPUT"    GRUPPO 2/2.NOME : ";N$(4)
600 GOSUB3390
610 PRINTCHR$(150)"■    BATTETE UN TASTO !"
620 GOSUB3390
630 GETC$:IFC$=""THEN630
640 PRINTCHR$(147):POKEV+21,0
650 PRINT:PRINT:PRINT,"GIOCA PRIMA :"
660 Z=8:S=10:GOSUB3500
670 PRINTN$(ZU):PRINT:PRINT,N$(ZU+1)
680 FORC=0TO5000:NEXT
690 :
700 REM..VIDEATA
710 :
720 PRINTCHR$(147):PRINTCHR$(5)
730 PRINT"   ┌─────────┐
740 PRINT"   |FATE LA|          ▁
750 PRINT"   |SCELTA |         /  \
760 IFZU=1THENPRINT"   |■GRUPPO1■|        /     \":REM INVERSO  ↑ & E
770 IFZU=3THENPRINT"   |■GRUPPO2■|        /     \":REM INVERSO  ↑ & E
780 PRINT"   └─────────┘       /  01  \
790 PRINT"                     \      /
800 PRINT"                      ▔▔▔▔
810 PRINTCHR$(129)
820 PRINT"                 ▁              ▁
830 PRINT"                /  \            /  \
840 PRINT"               /    \          /    \
850 PRINT"              /  02  \        /  03  \
860 PRINT"              \      /        \      /
870 PRINT"               ▔▔▔▔            ▔▔▔▔
880 PRINTCHR$(154)
890 PRINT"        ▁              ▁              ▁
900 PRINT"       /  \            /  \            /  \
910 PRINT"      /    \          /    \          /    \
920 PRINT"     /  04  \        /  05  \        /  06  \
930 PRINT"     \      /        \      /        \      /
940 PRINT"      ▔▔▔▔            ▔▔▔▔            ▔▔▔▔
```

```
950 PRINTCHR$(150)
960 Z=6:S=19:GOSUB3500
970 IFY(1)=1THEN PRINT "**"
980 Z=13:S=12:GOSUB3500
990 IFY(2)=1THEN PRINT "**"
1000 Z=13:S=27:GOSUB3500
1010 IFY(3)=1THEN PRINT "**"
1020 Z=20:S=19:GOSUB3500
1030 IFY(5)=1THEN PRINT "**"
1040 Z=20:S=33:GOSUB3500
1050 IFY(6)=1THEN PRINT "**"
1060 Z=20:S=6:GOSUB3500
1070 IFY(4)=1THEN PRINT "**"
1080 GOSUB3390
1090 GETC$:IFC$=""THEN1090
1100 IFASC(C$)<49ORASC(C$)>54THEN1090
1110 I=VAL(C$)
1120 IFY(I)=1THEN1090
1130 Y(I)=1:PRINTCHR$(147)
1140 IFZU=1THENPRINTCHR$(158)
1150 IFZU=3THENPRINTCHR$(153)
1160 :
1170 REM   1. ESAGONO
1180 :
1190 DATA 1. SPIAGGIA,SALVIETTA,SECCHIELLO,SOLE,VENTO,ONDE,CASTELLI
1200 DATA 2. GIARDINO,VERDE,PISCINA,PANCHINA,SIEPE,ALBERO,ERBA
1210 DATA 3. CUCINA,PANE,FORNELLO,CUOCERE,FARINA,PIATTO,SALAME
1220 DATA 4. INSETTI,CALABRONE,SCARABEO,VESPA,APE,MAGGIOLINO,MOSCA
1230 DATA 5. TECNICA,TRANSISTOR,MOTORE,PONTE,AEREOPLANO,AUTO,RADAR
1240 DATA 6. CHIMICA,ELETTRONE,ELEMENTO,ACIDO,FERRO,ATOMO,FUOCO
1250 :
1260 REM   2. ESAGONO
1270 :
1280 DATA 1. AUTO,TERGICRISTALLO,BATTERIA,CERCHIO,CHIAVE,VERNICE,TETTO
1290 DATA 2. ANIMALI ESOTICI,TIGRE,ZEBRA,SERPENTE,ELEFANTE,AIRONE,IENA
1300 DATA 3. LAGO/STAGNO,PALUDE,RANA,LIBELLULA,PESCE,ALGA,CANNA
1310 DATA 4. BEVANDE,BIRRA,LATTE,COKTAIL,BACARDI,VINO,SUCCO DI FRUTTA
1320 DATA 5. VERDURE,RABARBARO,CAVOLO,BARBABIETOLA,SPINACI,PISELLI,LENTICCHIE
1330 DATA 6. VACANZE,HOBBY,RIPOSO,GODIMENTO,VIAGGI,TEMPO LIBERO,RELAX
1340 :
1350 REM   3. ESAGONO
1360 :
1370 DATA 1. FIORI,CROCUS,TULIPANO,GAROFANO,ROSA,GIRASOLE,MARGHERITA
1380 DATA 2. FIUME,RUSCELLO,PORTO,SORGENTE,DEPURATORE,CASCATA,PONTE
1390 DATA 3. SALOTTO,LAMPADA,LAMPADARIO,QUADRO,LIBRERIA,TAPPETO,VIDEO
1400 DATA 4. LETTURA,VOCABOLO,GIORNALE,TITOLO,PAGINA,LIBRO,LETTERA
1410 DATA 5. FISICA,SUONO,MAGNETE,LUCE,MOTORE,CALORE,TEMPERATURA
1420 DATA 6. ABBIGLIAMENTO,BOTTONE,BIKINI,SCIALLE,GUANTO,CALZINO,GIACCONE
1430 :
1440 REM   4. ESAGONO
1450 :
1460 DATA1. SOTTOSUOLO,PETROLIO,FERRO,CARBONE,ORO,MARMO,METANO
1470 DATA2. UFFICIO,SCRIVANIA,MACCHINA,LIBRI,TELEFONO,OROLOGIO,LAVORO
1480 DATA3. CAMPEGGIO,TENDA,GRIGLIA,ROULOTTE,BRANDA,CAMPER,MATERASSINO
1490 DATA4. MUSICA,PIANOFORTE,NOTE,MELODIA,CORO,CANTO,FLAUTO
1500 DATA5. SPORT,TENNIS,CALCIO,GRAN PREMIO,RING,BOX,NUOTO
1510 DATA6. LAVORI,CAMERIERE,FORNAIO,CUOCO,REPORTER,MURATORE,IMPIEGATO
1520 :
1530 REM   5. ESAGONO
```

```
1540 :
1550 DATA 1. POSTA,TELEGRAMMI,TELEGRAFO,TIMBRO,PACCO,LETTERA,BOLLI
1560 DATA 2. CALCIO,RIGORE,PORTA,FUORI GIOCO,CAPITANO,GOAL,PORTA
1570 DATA 3. ROCK'N ROLL,HALEY,TURNER,COSTELLO,JAGGER,A.F.N.,ELVIS
1580 DATA 4. OSTERIA,CONTO,CARTE,BIRRA,BANCO,VINO,OSTE
1590 DATA 5. ROMA,COLLI,SAN PIETRO,COLOSSEO,TEVERE,SETTE RE,FORO
1600 DATA 6. PISCINA,ACCAPPATOIO,MAESTRO,TRAMPOLINO,ACQUA,PRATO,CABINA
1610 :
1620 REM   6. ESAGONO
1630 :
1640 DATA 1. ANIMALI DOMESTICI,CANE,GATTO,CAVIE,PASSERO,CRICETO,GUFO
1650 DATA 2. VELOCITA',JET,AEREO,AUTO,GHEPARDO,FULMINE,SECONDO
1660 DATA 3. FREDDO,INVERNO,NEVE,GHIAGGIO,GRANDINE,ARTIDE,CONGELATORE
1670 DATA 4. SCUOLA,CARTINA,DISEGNO,LIBRO,RIGA,COMPASSO,LAVAGNA
1680 DATA 5. FOTOGRAFIA,FLASH,STAMPA,CARTA,SVILUPPO,OBIETTIVO,SCHERMO
1690 DATA 6. CINEMA,TELONE,ATTORE,PROIETTORE,FILM,PROVINO,BIGLIETTO
1700 :
1710 REM.. VISUALIZZAZIONE ARGOMENTI
1720 :
1730 PRINTCHR$(147)
1740 FORF=1TO6
1750 EN=EN+1
1760 PRINTCHR$(147)
1770 G=ZA(1)+ZA(3)+ZA(5)
1780 H=ZA(2)+ZA(4)+ZA(6)
1790 P(ZU)=G:P(ZU+1)=H
1800 POKEV+21,0:POKE253,0
1810 PRINT:PRINT:PRINTTAB(1)LEFT$(N$(ZU),16);TAB(20)P(ZU)+S(ZU);"PUNTI
1820 PRINT:PRINTTAB(1)LEFT$(N$(ZU+1),16);TAB(20)P(ZU+1)+S(ZU+1);"PUNTI
1830 PRINTCHR$(5)
1840 PRINT:PRINT" SOMMA DEI PUNTI  : ";P(ZU)+P(ZU+1)+S(ZU)+S(ZU+1)
1850 IFZU=1THENPRINTCHR$(158)
1860 IFZU=3THENPRINTCHR$(153)
1870 PRINT:PRINT:PRINT
1880 FORC=1TO6
1890 Z=13:S=1:GOSUB3500
1900 IFI=CTHEN PRINT"ESAGONO ";C,
1910 NEXTC
1920 FORC=0TO10
1930 Z=10:S=1:GOSUB3500
1940 IFEN=C*4+1THEN PRINT LEFT$(N$(ZU+1),16);" SPIEGA I VOCABOLI!"
1950 IFEN=C*4+2THEN PRINT LEFT$(N$(ZU),16);" SPIEGA I VOCABOLI!"
1960 IFEN=C*4+3THEN PRINT LEFT$(N$(ZU+1),16);" SPIEGA I VOCABOLI!"
1970 IFEN=C*4+4THEN PRINT LEFT$(N$(ZU),16);" SPIEGA I VOCABOLI!"
1980 NEXTC
1990 Z=15:S=0:GOSUB3500
2000 FORB=(I*42-41)TO(I*42)STEP7:PRINT"   "A$(B):GOSUB3390:NEXTB
2010 PRINT:PRINT" SCEGLIETE UN ARGOMENTO (1-6) !"
2020 FORC=1TO6
2030 IFX(C)=1THENZ=14+C:S=0:GOSUB3500
2040 IFX(C)=1THENPRINT"—>"
2050 NEXTC
2060 GETC$:IFC$=""THEN2060
2070 PY=VAL(C$)
2080 IFASC(C$)<49ORASC(C$)>54THEN2060
2090 C=VAL(C$)
2100 IFX(C)=1THENGOTO2060
2110 X(C)=1:PRINTCHR$(147)
2120 TI$="000000"
```

```
2130 FORE=0TO5
2140 GOSUB2580
2150 NEXTE
2160 SS=0:NEXTF
2170 :
2180 REM..RISULTATO FINALE
2190 :
2200 G=ZA(1)+ZA(3)+ZA(5)
2210 H=ZA(2)+ZA(4)+ZA(6)
2220 IFZU=1GOTO2930
2230 IFZU=3GOTO3000
2240 PRINTCHR$(147):PRINTCHR$(5)
2250 POKEV+21,0:POKE253,0
2260 PRINT"      ****  R I S U L T A T O ****"
2270 HG=12*4096:REM INDIRIZZO GRAFICA
2280 PRINT:PRINT:PRINTLEFT$(N$(1),11);TAB(12)S(1)
2290 Z=5:S=17:GOSUB3500
2300 SYSHG,S(1)*1.7,15
2310 PRINT:PRINTLEFT$(N$(2),11);TAB(12)S(2)
2320 Z=7:S=17:GOSUB3500
2330 SYSHG,S(2)*1.7,15
2340 PRINTCHR$(158)
2350 PRINT:PRINT"GRUPPO 1    :";TAB(12)S(1)+S(2)
2360 Z=10:S=17:GOSUB3500
2370 SYSHG,(S(2)+S(1))*1.7,7
2380 PRINTCHR$(5)
2390 PRINT"- - - - - - - - - - - - - - - - - - - -"
2400 PRINTCHR$(30)
2410 PRINT"GRUPPO 2    :";TAB(12)S(3)+S(4)
2420 PRINTCHR$(5)
2430 Z=14:S=17:GOSUB3500
2440 SYSHG,(S(3)+S(4))*1.7,5
2450 PRINT:PRINT:PRINTLEFT$(N$(3),11);TAB(12)S(3)
2460 Z=17:S=17:GOSUB3500
2470 SYSHG,S(3)*1.7,12
2480 PRINT:PRINT LEFT$(N$(4),11);TAB(12)S(4)
2490 Z=19:S=17:GOSUB3500
2500 SYSHG,S(4)*1.7,12
2510 PRINT:PRINT:PRINT"PREMETE LA BARRA DELLO SPAZIO!"
2520 GETC$:IFC$=""THEN2520
2530 IFASC(C$)<>32 THEN2520
2540 FORP=1TO6:X(P)=0:NEXT
2550 IFY(1)=1ANDY(2)=1ANDY(3)=1ANDY(4)=1ANDY(5)=1ANDY(6)=1THENGOTO2860
2560 GOTO700
2570 :
2580 REM..TOTALIZZATORE
2590 :
2600 GOSUB3920
2610 Z=20:S=10:GOSUB3500
2620 PRINT"IL TEMPO PASSA....."
2630 Z=10:S=10:GOSUB3500
2640 SYS828:POKE251,150:POKE252,165:POKE253,15:POKE254,0
2650 PRINTA$((C*7-6)+(I-1)*42)
2660 Z=13:S=10:GOSUB3500
2670 PRINT"                                    "
2680 Z=13:S=10:GOSUB3500
2690 PRINTCHR$(144)A$((PY*7-E)+(I-1)*42)
2700 PRINTCHR$(158)
2710 :
```

```
2720 :
2730 AA=INT(TI/60):REM TEMPO A DISPOSIZIONE
2740 :
2750 :
2760 IFAA>39THENGOSUB5610:RETURN
2770 IFAA>39THENRETURN
2780 FORK=AATO39:POKE1944+K,81:NEXTK
2790 FORK=AATO39:POKE56216+K,14:NEXTK
2800 FORP=1TOAA:POKE1943+P,32:NEXTP
2810 GETB$:IFB$<>"S"ANDB$<>"N"THEN2730
2820 IFB$="N"THEN2840
2830 ZA(F)=ZA(F)+1:GOSUB3390
2840 RETURN
2850 :
2860 REM..
2870 :
2880 Z=22:S=0 :GOSUB3500
2890 :
2900 PRINT"           ■*** FINE GIOCA! ***■",,"ANCORA        -> RUNSTOP,RUN !'
2910 GOTO2910:REM RIGA 2900 = INVERSE ATT/DISATTIVA
2920 :
2930 REM..PUNTI  G1
2940 :
2950 S(1)=S(1)+G
2960 S(2)=S(2)+H
2970 ZU=3
2980 FORC=0TO6:ZA(C)=0:NEXT
2990 GOTO2240
3000 REM..PUNTI  G2
3010 S(3)=S(3)+G
3020 S(4)=S(4)+H
3030 ZU=1
3040 FORC=0TO6:ZA(C)=0:NEXT
3050 GOTO2240
3060 :
3070 REM..FINE
3080 :
3090 DIMA$(252)
3100 FORN=1TO252
3110 READZ$:A$(N)=Z$
3120 NEXTN
3130 REM....DIAGRAMMA MC
3140 :
3150 FORC=49152TO49359
3160 READX:POKEC,X:NEXT
3170 DATA32,253,174,32,235,183,134,36,165,21,201,2
3180 DATA176,66,10,10,10,10,10,133,35,165,20,72
3190 DATA74,74,74,24,101,35,101,211,72,168,197,211
3200 DATA240,19,201,39,144,2,160,39,32,36,234,169
3210 DATA160,32,30,234,136,196,211,16,243,104,168,104
3220 DATA192,40,176,11,41,7,170,189,83,192,166,36
3230 DATA32,30,234,169,17,76,22,231,76,72,178,32
3240 DATA101,116,117,97,246,234,231,32,253,174,32,235
3250 DATA183,165,21,208,235,134,36,165,20,74,74,74
3260 DATA133,35,165,20,41,7,133,37,165,209,24,101
3270 DATA211,133,253,165,210,105,0,133,254,160,0,166
3280 DATA35,240,32,32,199,192,169,160,145,253,165,36
3290 DATA145,243,165,253,56,233,40,133,253,176,8,198
3300 DATA254,165,254,201,4,144,18,198,35,208,224,32
```

```
3310 DATA199,192,166,37,189,191,192,145,253,165,36,145
3320 DATA243,165,211,197,213,240,3,76,59,171,96,32
3330 DATA100,111,121,98,248,247,227,165,253,133,243,165
3340 DATA254,76,42,234
3350 RETURN
3360 :
3370 REM..SUONI
3380 :
3390 SI=54272:FL=SI:FH=SI+1
3400 W=SI+4:A=SI+5:H=SI+6:L=SI+24
3410 POKEL,15:POKEA,9:POKEA+7,9
3420 POKEH,0:POKEH+7,0
3430 POKEFH,25:POKEFL,255
3440 POKEFH+7,30:POKEFL+7,255
3450 POKEW,17:POKEW+7,17
3460 FORT=0TO500:NEXT
3470 POKEW,0:POKEA,0:POKEH,0:REM FINE
3480 POKEW+7,0:POKEA+7,0:POKEH+7,0:REM AUS
3490 RETURN
3500 :
3510 REM....CURSORE
3520 :
3530 POKE214,Z:POKE211,S:SYS58640
3540 RETURN
3550 :
3560 REM SPRITES 0 - 6
3570 :
3580 V=53248:SO=2040:PO=33
3590 POKEV+21,127:POKEV+27,127
3600 FORI=0TO6:POKESO+I,PO+I:NEXT
3610 FORJ=0TO6
3620 FORI=0TO62:READQ:POKE2112+I+(J*64),Q:NEXTI:NEXTJ
3630 Z=18:S=0:GOSUB3500
3640 FORX=0TO231
3650 POKEV+12,X  :POKEV+13,90:NEXT
3660 FORX=0TO208
3670 POKEV+10,X  :POKEV+11,90:NEXT
3680 FORX=0TO184
3690 POKEV+8,X   :POKEV+9,90 :NEXT
3700 FORX=0TO161
3710 POKEV+6,X   :POKEV+7,90 :NEXT
3720 FORX=0TO138
3730 POKEV+4,X   :POKEV+5,90 :NEXT
3740 FORX=0TO115
3750 POKEV+2,X   :POKEV+3,90 :NEXT
3760 FORX=0TO93
3770 POKEV,X   :POKEV+1,90 :NEXT
3780 PRINT"(C) TECNICHE NUOVE "
3790 PRINT"VIA MOSCOVA  46/9A "
3800 POKEV+39,INT(RND(1)*14)+1
3810 POKEV+40,INT(RND(1)*14)+1
3820 POKEV+41,INT(RND(1)*14)+1
3830 POKEV+42,INT(RND(1)*14)+1
3840 POKEV+43,INT(RND(1)*14)+1
3850 POKEV+44,INT(RND(1)*14)+1
3860 POKEV+45,INT(RND(1)*14)+1
3870 POKEV+46,INT(RND(1)*14)+1
3880 POKEV+47,INT(RND(1)*14)+1
```

```
3890 ZX=ZX+1
3900 IFZX<25THENGOTO3800
3910 GOTO400
3920 POKEV+21,127:FORXY=0TO6:POKEV+39+XY,INT(RND(1)*14)+1:NEXT
3930 RETURN
3940 :
3950 :
3960 REM DATI PER SPRITE
3970 :
3980 :
3990 DATA 3, 192, 15
4000 DATA 3, 192, 15
4010 DATA 3, 192, 15
4020 DATA 7, 128, 30
4030 DATA 7, 128, 30
4040 DATA 7, 128, 30
4050 DATA 15, 0, 60
4060 DATA 15, 0, 60
4070 DATA 15, 0, 60
4080 DATA 31, 255, 248
4090 DATA 31, 255, 248
4100 DATA 31, 255, 248
4110 DATA 60, 0, 240
4120 DATA 60, 0, 240
4130 DATA 60, 0, 240
4140 DATA 120, 1, 224
4150 DATA 120, 1, 224
4160 DATA 120, 1, 224
4170 DATA 240, 3, 192
4180 DATA 240, 3, 192
4190 DATA 240, 3, 192
4200 DATA 3, 255, 255
4210 DATA 3, 255, 255
4220 DATA 3, 255, 254
4230 DATA 7, 128, 14
4240 DATA 7, 128, 0
4250 DATA 7, 128, 0
4260 DATA 15, 0, 0
4270 DATA 15, 0, 0
4280 DATA 15, 0, 0
4290 DATA 31, 254, 0
4300 DATA 31, 254, 0
4310 DATA 31, 252, 0
4320 DATA 60, 0, 0
4330 DATA 60, 0, 0
4340 DATA 60, 0, 0
4350 DATA 120, 0, 0
4360 DATA 120, 0, 0
4370 DATA 120, 1, 192
4380 DATA 255, 255, 192
4390 DATA 255, 255, 192
4400 DATA 255, 255, 128
4410 DATA 3, 192, 15
4420 DATA 3, 192, 15
4430 DATA 1, 224, 30
4440 DATA 1, 224, 60
4450 DATA 1, 224, 120
4460 DATA 0, 240, 240
4470 DATA 0, 241, 224
```

```
4480 DATA 0, 243, 192
4490 DATA 0, 127, 128
4500 DATA 0, 127, 0
4510 DATA 0, 62, 0
4520 DATA 0, 126, 0
4530 DATA 0, 254, 0
4540 DATA 1, 239, 0
4550 DATA 3, 207, 0
4560 DATA 7, 135, 128
4570 DATA 15, 7, 128
4580 DATA 30, 7, 128
4590 DATA 60, 3, 192
4600 DATA 120, 3, 192
4610 DATA 240, 3, 192
4620 DATA 1, 255, 254
4630 DATA 3, 255, 255
4640 DATA 3, 255, 255
4650 DATA 7, 128, 30
4660 DATA 7, 128, 30
4670 DATA 7, 128, 30
4680 DATA 15, 0, 60
4690 DATA 15, 0, 60
4700 DATA 15, 0, 60
4710 DATA 31, 255, 248
4720 DATA 31, 255, 248
4730 DATA 31, 255, 248
4740 DATA 60, 0, 240
4750 DATA 60, 0, 240
4760 DATA 60, 0, 240
4770 DATA 120, 1, 224
4780 DATA 120, 1, 224
4790 DATA 120, 1, 224
4800 DATA 240, 3, 192
4810 DATA 240, 3, 192
4820 DATA 240, 3, 192
4830 DATA 1, 255, 254
4840 DATA 3, 255, 255
4850 DATA 3, 255, 255
4860 DATA 7, 128, 30
4870 DATA 7, 128, 30
4880 DATA 7, 128, 30
4890 DATA 15, 0, 0
4900 DATA 15, 0, 0
4910 DATA 15, 0, 0
4920 DATA 30, 0, 0
4930 DATA 30, 7, 248
4940 DATA 30, 15, 248
4950 DATA 60, 15, 240
4960 DATA 60, 0, 240
4970 DATA 60, 0, 240
4980 DATA 120, 1, 224
4990 DATA 120, 1, 224
5000 DATA 120, 1, 224
5010 DATA 255, 255, 192
5020 DATA 255, 255, 192
5030 DATA 127, 255, 128
5040 DATA 1, 255, 254
5050 DATA 3, 255, 255
5060 DATA 3, 255, 255
```

```
5070 DATA 7, 192, 62
5080 DATA 7, 128, 30
5090 DATA 7, 128, 30
5100 DATA 15, 0, 60
5110 DATA 15, 0, 60
5120 DATA 15, 0, 60
5130 DATA 30, 0, 120
5140 DATA 30, 0, 120
5150 DATA 30, 0, 120
5160 DATA 60, 0, 240
5170 DATA 60, 0, 240
5180 DATA 60, 0, 240
5190 DATA 120, 1, 224
5200 DATA 120, 1, 224
5210 DATA 124, 3, 224
5220 DATA 255, 255, 192
5230 DATA 255, 255, 192
5240 DATA 127, 255, 128
5250 DATA 3, 192, 15
5260 DATA 3, 192, 15
5270 DATA 3, 224, 15
5280 DATA 7, 224, 30
5290 DATA 7, 224, 30
5300 DATA 7, 240, 30
5310 DATA 15, 112, 60
5320 DATA 15, 112, 60
5330 DATA 15, 56, 60
5340 DATA 30, 56, 120
5350 DATA 30, 56, 120
5360 DATA 30, 28, 120
5370 DATA 60, 28, 240
5380 DATA 60, 28, 240
5390 DATA 60, 14, 240
5400 DATA 120, 15, 224
5410 DATA 120, 7, 224
5420 DATA 120, 7, 224
5430 DATA 240, 3, 192
5440 DATA 240, 3, 192
5450 DATA 240, 3, 192
5460 :
5470 REM COLONNE/COLORI
5480 :
5490 FORI=828TO911
5500 READX:POKEI,X:S=S+X:NEXTI
5510 DATA120,169,91,141,20,3,169,3,141,21,3,165
5520 DATA251,141,18,208,173,17,208,41,127,141,17,208
5530 DATA169,129,141,26,208,88,96,173,25,208,141,25
5540 DATA208,41,1,208,7,173,13,220,88,76,49,234
5550 DATA173,18,208,197,252,176,13,165,253,141,32,208
5560 DATA141,33,208,165,252,76,138,3,165,254,141,32
5570 DATA208,141,33,208,165,251,141,18,208,76,188,254,
5580 RETURN
5590 :
5600 :
5610 REM TIRO
5620 :
5630 :
5640 IFSS>0THEN5700
5650 SI=54272:FL=SI+1:TL=SI+2:TH=SI+3:W=SI+4:A=SI+5:H=SI+6:L=SI+24
```

```
5660 FORX=15 TO 0 STEP-1
5670 POKEL,X:POKEA,15:POKEH,0:POKEFH,40:POKEFL,200:POKEW,129
5680 NEXT
5690 POKEW,0:POKEA,0
5700 SS=SS+1
5710 RETURN
```

Casinò-Blackjack

Lunghezza del programma: 7170 byte. Linguaggio di programmazione: BASIC. Accessori: nessuno. Giocatori: 1.

Il gioco d'azzardo "Blackjack" è una variante di un gioco di origine popolare, "17 e 4". Questo programma originariamente è stato scritto per escogitare delle strategie di gioco per il blackjack che evitassero gravi perdite finanziarie. Perciò si è data molta importanza a riprodurre con esattezza le condizioni e le regole di gioco dei casinò.
Dopo l'introduzione e il lancio del programma, sullo schermo appare il tipico tavolo da blackjack. Voi giocate nella casella centrale, quella leggermente sfalsata rispetto alle altre, e potete ignorare tranquillamente quello che succede nelle tre caselle a sinistra e nelle tre caselle a destra. Per simulare un tavolo completo infatti, abbiamo inserito sei giocatori fittizi. Quando nella vostra casella appare il punto di domanda, dovete fare la vostra puntata, che deve essere un valore multiplo di dieci, compreso nell'intervallo tra 10 e 500. All'inizio del gioco avete un capitale iniziale di 0. Quando confermate la puntata con RETURN, vengono date le carte e ogni volta vi viene chiesto se ne volete un'altra. Se volete una carta premete il tasto (C), "carta", altrimenti il tasto (P), "passo". Il croupier, dopo aver interrogato tutti i giocatori, si dà le carte e poi inizia il conteggio dei punti secondo le regole del gioco.
Vi viene chiesto anche se volete assicurarvi contro un blackjack, se volete aumentare la posta, ecc.; per rispondere premete i tasti "S" (per sì) o "N" (per no).
Con questo programma, a differenza di quanto avviene con i programmi più economici, farete delle partite a blackjack che offrono le stesse condizioni e le stesse regole dei casinò.

Thomas Hansen

```
1 REM********************
2 REM*"TH-BLACK-JACK1.2"*
3 REM*  EIN PRODUKT VON *
4 REM*   THOMAS HANSEN  *
5 REM*WEICHSELSTRASSE 10*
6 REM* 2370 BUEDELSDORF *
7 REM********************
80 S=0:FORI=2048TO2230:S=S+PEEK(I):NEXT
90 IFS<>9671THENPRINT"ERRORE DI LETTURA NELLE POKE!":END
100 DIMK$(207)
110 DIMKC$(10):DIMKS$(10,1)
120 SKF$(0)=" ╭──╮ "
130 SKF$(1)="|    |"
140 SKF$(2)=" ╰──╯ "
150 POKE53280,0:POKE53281,5
160 POKE53272,PEEK(53272)AND253:Q=(PEEK(2136)+5)*(PEEK(2143)-48):
    PRINTCHR$(8)
170 N=208
180 PRINT"■",,,"     ▄▄▄▄▄▄▄▄ ";
190 PRINT,,,"   ▌        ▐ ";
200 PRINT,,,"   ▌        ▐ ";
210 PRINT,,,"   ▌        ▐ ";
220 PRINT,,,"   ▌        ▐ ";
```

```
230 PRINT,,,"    ■■■■■■■■";
240 PRINT"■■■■■■■■"
250 PRINT"          ■BLACK JACK PAYS 3 TO 2
260 PRINT"■DEALER MUST STAND ON 17 AND DRAW ON 16
270 PRINT"■  ┌──────────────────────────────┐
280 PRINT"  |     INSURANCE  PAYS 2 TO 1     |
290 PRINT"  └──────────────────────────────┘
300 PRINT"  ┌──┐ ┌──┐ ┌──┐   ┌──┐   ┌──┐ ┌──┐ ┌──┐"
310 PRINT"  |■10■||■10■||■10■|   |  |   |■10■||■10■||■10■|"
320 PRINT"  └──┘ └──┘ └──┘   └──┘   └──┘ └──┘ └──┘"
330 GOSUB2060:GOSUB2240
340 Z=22:S=16:GOSUB2230:PRINT"■"E(0)
350 Z=24:S=15:GOSUB2230:PRINTSPC(5-LEN(STR$(B)))B;
360 POKE198,0
370 Z=22:S=15:GOSUB2230:INPUTEZ$
380 E(0)=VAL(LEFT$(EZ$,2)):IFMID$(EZ$,3,1)<>"|"THENE(0)=
    VAL(LEFT$(EZ$,3))
390 IFE(0)<10ORE(0)>500OR(E(0)/10-INT(E(0)/10))<>0THEN370
400 IFE(0)<100THENZ=22:S=16:GOSUB2230:PRINTE(0)"■■|■■■■■■ |"
410 IFE(0)>=100THENZ=22:S=16:GOSUB2230:PRINTE(0)"■■■■■■■■ |"
420 B=B-E(0):Z=24:S=15:GOSUB2230:PRINT"■"SPC(5-LEN(STR$(B)))B;
430 IFN<175THEN460
440 LS=1:Z=1:S=1:GOSUB2230
450 PRINT"■ULTIMO ■■■■■■■■GIOCO    ■■■■■■■■■■■DI    ■■■■■■■■OGGI!"
460 CN=0:SN(0)=0:SN(1)=0
470 FORI=0TO6:MN(I)=0:MS(I)=0:NEXTI
480 FORII=0TO2:GOSUB2350:NEXTII
490 G=0:GOSUB1600
500 FORII=4TO6:GOSUB2350:NEXTII
510 G=2:UD=0:GOSUB1600
520 Z=1:S=1:GOSUB2230
530 PRINT"      ■■■■■■■■■■     ■■■■■■■■■■■     ■■■■■■■■        "
540 INS=0
550 IFRIGHT$(KC$(0),1)<>"A"THEN660
560 Z=1:S=1:GOSUB2230
570 PRINT"■VOLETE    ■■■■■■■■■■■■ASSICURA-■■■■■■■■■■■■RVI?"
580 GETA$:IFA$=""THEN580
590 IFA$="N"THEN640
600 INS=1
610 B=B-E(0)/2
620 Z=24:S=15:GOSUB2230:PRINT"■"SPC(5-LEN(STR$(B)))B;
630 Z=19:S=16:GOSUB2230:PRINTE(0)/2"■■ "
640 Z=1:S=1:GOSUB2230
650 PRINT"      ■■■■■■■■■■■■■     ■■■■■■■■■■■■       "
660 FORII=0TO2:GOSUB2350:NEXTII
670 G=0:GOSUB1600
680 FORII=4TO6:GOSUB2350:NEXTII
690 G=2:UD=1:GOSUB1600
700 SPLITT=0
710 MAX(0)=100:MAX(1)=100
720 IFRIGHT$(KS$(0,0),1)<>RIGHT$(KS$(1,0),1)THEN850
730 Z=1:S=1:GOSUB2230
740 PRINT"■VOLETE    ■■■■■■■■■■■■■SCARTARE?"
750 GETA$:IFA$<>"S"ANDA$<>"Y"ANDA$<>"N"THEN750
760 IFA$="N"THEN830
770 SPLITT=1
780 E(1)=E(0):B=B-E(0)
790 Z=24:S=15:GOSUB2230:PRINT"■"SPC(5-LEN(STR$(B)))B;
800 KS$(0,1)=KS$(1,0)
```

```
810  SN(0)=0:SN(1)=0
820  IFRIGHT$(KS$(0,0),1)="A"ANDRIGHT$(KS$(0,1),1)="A"THENMAX(0)
     =2:MAX(1)=2
830  Z=1:S=1:GOSUB2230
840  PRINT"               ■■■■■■■■■■■■■                 "
850  FORII=0TO2:GOSUB2350:NEXTII
860  FORGAME=0TOSPLITT
870  IFSPLITT=0ORSN(GAME)<>0THEN960
880  FORI=0TOSN(0)+1
890  Z=13-2*I:S=16+I:GOSUB2230
900  PRINT"     ■■■■■     ■■■■■     ■■■■■     ■■■■■     ";
910  NEXTI
920  G=GAME:GOSUB1860
930  G=GAME:GOSUB1600
940  IFE(0)<100THENZ=22:S=16:GOSUB2230:PRINT"■"E(0)"■■|■■■■■■ |"
950  IFE(0)>=100THENZ=22:S=16:GOSUB2230:PRINT"■"E(0)"■■■■■■■■ |"
960  IFSN(GAME)<>2ORPS(GAME)<9ORPS(GAME)>11THEN1080
970  Z=1:S=1:GOSUB2230
980  PRINT"■VOLETE    ■■■■■■■■■■■■RADDOPPIARE?"
990  GETA$:IFA$<>"S"ANDA$<>"Y"ANDA$<>"N"THEN990
1000  IFA$="N"THEN1060
1010  B=B-E(GAME):E(GAME)=2*E(GAME)
1020  IFE(GAME)<100THENZ=22:S=16:GOSUB2230:PRINT"■"E(0)"■■|■■■■■■ |"
1030  IFE(GAME)>=100THENZ=22:S=16:GOSUB2230:PRINT"■"E(0)"■■■■■■■■ |"
1040  Z=24:S=15:GOSUB2230:PRINT"■"SPC(5-LEN(STR$(B)))B;
1050  MAX(GAME)=3
1060  Z=1:S=1:GOSUB2230
1070  PRINT"               ■■■■■■■■■■■■■          ■|"
1080  G=GAME
1090  IFPS(G)>20THEN1210
1100  IFMAX(GAME)=3THEN1170
1110  IFMAX(GAME)<>2THEN1140
1120  FORI=1TO2000:NEXT
1130  GOTO1210
1140  Z=1:S=1:GOSUB2230:PRINT"■CARTA?";
1150  GETA$:IFA$<>"C"ANDA$<>"P"THEN1150
1160  IFA$="P"THEN1210
1170  GOSUB1600
1180  IFGAME=0THENFORII=4TO6:GOSUB2350:NEXTII
1190  IFGAME=0THENFORII=0TO2:GOSUB2350:NEXTII
1200  IFPS(G)<21ANDMAX(GAME)>SN(GAME)THENGOTO1080
1210  Z=1:S=1:GOSUB2230:PRINT"          ";
1220  FORII=4TO6:GOSUB2350:NEXTII
1230  FORIJ=1TO8
1240  FORII=0TO2:GOSUB2350:NEXTII
1250  FORII=4TO6:GOSUB2350:NEXTII
1260  NEXTIJ
1270  NEXTGAME
1280  G=2:UD=2:GOSUB1600
1290  IFPC<17THEN1280
1300  IFPC<22THEN1330
1310  Z=1:S=1:GOSUB2230
1320  PRINT"■LA BANCA■■■■■■■■■■HA          ■■■■■■■■■■CHIUSO!"
1330  BC=0:IFCN=2ANDPC=21THENBC=1
1340  FORI=0TO1
1350  BS(I)=0:IFSN(I)=2ANDPS(I)=21ANDSPLITT=0THENBS(I)=1
1360  NEXT
1370  FORGAME=0TOSPLITT
```

```
1380 IFINS=0THEN1430
1390 IFBC=0THEN1430
1400 REM INSURANCE
1410 GE=1.5*E(GAME)
1420 B=B+GE
1430 IFPS(GAME)<22AND(PS(GAME)>=PCORPC>21)AND(BC=0ORBS(GAME)=1)THEN1470
1440 REM PERSO
1450 GE=0
1460 GOTO1560
1470 IFPS(GAME)>PCORPC>21OR(BS(GAME)=1ANDBC=0)THEN1520
1480 REM NON DECISIVO
1490 GE=E(GAME)
1500 B=B+GE
1510 GOTO1560
1520 GE=2*E(GAME)
1530 IFBS(GAME)=1ANDBC=0THENGE=GE+E(GAME)/2
1540 REM VINTO
1550 B=B+GE
1560 NEXTGAME
1570 IFLS=1THENN=208
1580 FORI=1TO5000:NEXT
1590 GOTO180
1600 IFN=208THENGOSUB2520
1610 IFG<2THEN1850
1620 IFUD=2ANDCN=1THEN1640
1630 KC$(CN)=K$(N)
1640 Z=YC:S=3*CN+XC:GOSUB2230:PRINT"■"SKF$(0);
1650 FORI=1TO3:PRINT"■■■■■"SKF$(1);:NEXT
1660 PRINT"■■■■■"SKF$(2);
1670 IFLEFT$(KC$(CN),1)="♣"ORLEFT$(KC$(CN),1)="♠"THENPRINT"■";:
     GOTO1690
1680 PRINT"■";
1690 IFUD<>1THEN1720
1700 PRINT"□□□■■■■■■■■■■■■■■■"
1710 GOTO2050
1720 PRINT"□□□■■"MID$(KC$(CN),2,2)
1730 Z=2:S=3*CN+12:GOSUB2230:PRINTLEFT$(KC$(CN),1)"■"
1740 PC=0:AC=0
1750 FORI=0TOCN
1760 IFASC(RIGHT$(KC$(I),1))<58THENPC=PC+VAL(MID$(KC$(I),2,2))
1770 IFASC(RIGHT$(KC$(I),1))>64THENPC=PC+10
1780 IFRIGHT$(KC$(I),1)="A"THENPC=PC+1:AC=AC+1
1790 NEXT
1800 IFPC<22ORAC=0THEN1820
1810 PC=PC-10:AC=AC-1:GOTO1800
1820 CN=CN+1
1830 IFCN<>2THEN2050
1840 RETURN
1850 KS$(SN(G),G)=K$(N)
1860 Z=YS-2*SN(G):S=XS+SN(G):GOSUB2230:PRINT"■"SKF$(0);
1870 FORI=1TO3:PRINT"■■■■■"SKF$(1);:NEXT
1880 PRINT"■■■■■"SKF$(2);
1890 IFLEFT$(KS$(SN(G),G),1)<>"♣"ANDLEFT$(KS$(SN(G),G),1)<>"♠"
     THENGOTO1910
1900 PRINT"■";:GOTO1920
1910 PRINT"■";
1920 PRINT"□□□■■"MID$(KS$(SN(G),G),2,2)
1930 Z=YT-2*SN(G):S=XT+SN(G):GOSUB2230:PRINTLEFT$(KS$(SN(G),G),1);
```

```
1940 IFMID$(KS$(SN(G),G),2,2)="10"THENPRINT"■■■"MID$(KS$(SN(G),G),2,2)"■";
1950 IFMID$(KS$(SN(G),G),2,2)<>"10"THENPRINT"■"MID$(KS$(SN(G),G),2,1)"■";
1960 PS(G)=0:AS(G)=0
1970 FORI=0TOSN(G)
1980 IFASC(RIGHT$(KS$(I,G),1))<58THENPS(G)=PS(G)+VAL(MID$(KS$(I,G),2,2))
1990 IFASC(RIGHT$(KS$(I,G),1))>64THENPS(G)=PS(G)+10
2000 IFRIGHT$(KS$(I,G),1)="A"THENPS(G)=PS(G)+1:AS(G)=AS(G)+1
2010 NEXT
2020 IFPS(G)<22ORAS(G)=0THEN2040
2030 PS(G)=PS(G)-10:AS(G)=AS(G)-1:GOTO2020
2040 SN(G)=SN(G)+1
2050 N=N+1
2060 I1=INT((211-N)/35)
2070 I2=INT((211-N)*8/35)-8*I1
2080 SD$=""
2090 FORI=6TOI1+SGN(I2)STEP-1:SD$=SD$+"  ":NEXTI
2100 IFI2=7THENSD$=SD$+"■ ■"
2110 IFI2=6THENSD$=SD$+"■ ■"
2120 IFI2=5THENSD$=SD$+"■ ■"
2130 IFI2=4THENSD$=SD$+"■ ■"
2140 IFI2=3THENSD$=SD$+" ▌"
2150 IFI2=2THENSD$=SD$+" ▌"
2160 IFI2=1THENSD$=SD$+" ▏"
2170 IFI1>1THENFORI=1TOI1-1:SD$=SD$+"■ ■":NEXTI
2180 IFI1>0THENSD$=SD$+"■ ■"
2190 FORI=1TO4
2200 Z=I:S=32:GOSUB2230:PRINTSD$;
2210 NEXTI
2220 RETURN
2230 POKE211,S:POKE214,Z:SYS58732:RETURN
2240 YC=PEEK(Q+05)-PEEK(Q+10)
2250 XC=PEEK(Q+05)-PEEK(Q+07)
2260 YS=PEEK(Q+12)-PEEK(Q+04)
2270 XS=PEEK(Q+10)-PEEK(Q+26)
2280 YT=PEEK(Q+00)-PEEK(Q+11)
2290 XT=PEEK(Q+00)-PEEK(Q+26)
2300 YG=PEEK(Q+09)-PEEK(Q+04)
2310 XG=PEEK(Q+02)-PEEK(Q+03)
2320 YH=PEEK(Q+00)-PEEK(Q+24)
2330 XH=PEEK(Q+07)-PEEK(Q+11)
2340 RETURN
2350 IFN=208THENGOSUB2520
2360 IFMS(II)>16THENRETURN
2370 Z=YG-2*MN(II):S=4*II+4*INT(II/4)+MN(II)+XG:GOSUB2230:
     PRINT"■"SKF$(0);
2380 FORI=1TO3:PRINT"■■■■■"SKF$(1);:NEXT
2390 PRINT"■■■■■"SKF$(2);
2400 IFLEFT$(K$(N),1)<>"♣"ANDLEFT$(K$(N),1)<>"♠"THENGOTO2420
2410 PRINT"■";:GOTO2430
2420 PRINT"■";
2430 PRINT"□□□■■■"MID$(K$(N),2,2)
2440 Z=YH-2*MN(II):S=4*II+4*INT(II/4)+MN(II)+XH:GOSUB2230:
     PRINTLEFT$(K$(N),1);
2450 IFMID$(K$(N),2,2)="10"THENPRINT"■■■"MID$(K$(N),2,2)"■";
2460 IFMID$(K$(N),2,2)<>"10"THENPRINT"■"MID$(K$(N),2,1)"■";
2470 IFASC(RIGHT$(K$(N),1))<58THENMS(II)=MS(II)+VAL(MID$(K$(N),2,2))
2480 IFASC(RIGHT$(K$(N),1))>64THENMS(II)=MS(II)+10
2490 IFRIGHT$(K$(N),1)="A"THENMS(II)=MS(II)+1
2500 MN(II)=MN(II)+1
```

```
2510 GOTO2050
2520 REM MESCOLARE
2530 Z=1:S=1:GOSUB2230
2540 PRINT"■MISCHIO    ■■■■■■■■■■■■LE          ■■■■■■■■■■■■CARTE.■";
2550 PRINT"■■■■■■■■■          ";
2560 N=0
2570 FORI1=1TO4
2580 FORI2=1TO4
2590 IFI2=1THENF$="♣"
2600 IFI2=2THENF$="♠"
2610 IFI2=3THENF$="♥"
2620 IFI2=4THENF$="♦"
2630 RESTORE
2640 FORI3=NTON+12
2650 READW$
2660 K$(I3)=F$+W$
2670 NEXTI3
2680 N=N+13
2690 NEXTI2
2700 NEXTI1
2710 DATA2,3,4,5,6,7,8,9,10,J,Q,K,A
2720 R=RND(-TI)
2730 FORI=0TO207
2740 R=INT(RND(0)*208)
2750 F$=K$(R)
2760 K$(R)=K$(I)
2770 K$(I)=F$
2780 NEXT
2790 Z=1:S=1:GOSUB2230:
     PRINT"          ■■■■■■■■■■          ■■■■■■■■■■          ";
2800 N=0
2810 LS=0
2820 RETURN
```

La terra dei 7 anelli

Lunghezza del programma: 19855 byte. Linguaggio di programmazione: BASIC. Accessori: registratore.

Vi trovate in un bosco secolare, nelle vicinanze del leggendario castello di "Canterville". Questo castello, infestato da spiriti maligni, può essere liberato dalla sua maledizione grazie a "sette anelli"; la condizione è che le forze del male siano prima confinate in un carcere sotterraneo. Per riuscire in questa impresa avete a disposizione tre vite; se ne perdete una, dovete ripartire dall'inizio e tutti gli oggetti che vi eravate conquistati, vanno perduti. Se non eravate ancora riusciti ad entrare nel castello, ripartire dall'inizio non è un male.
Per spostarvi dovete battere la lettera iniziale della direzione in cui volete andare, per esempio "N" per Nord o "A" per alto. Esistono anche altri comandi che si utilizzano da soli o in riferimento a degli oggetti (+G).
"L" ("look around"): questo comando offre una descrizione della situazione in cui si trova il giocatore.
"I" ("inventory"): il computer dà un elenco degli oggetti conquistati.
"GRIDA": quando si usa questo comando solitamente non succede nulla. Attenzione!
"DOMANDA": è molto utile domandare alle persone che cosa vogliono.
"APRITI": per esempio per le porte segrete ...
"UCCIDI": questo comando è necessario nella lotta contro i mostri che si incontrano via via.
"DAI" + O: anche corrompere qualcuno può essere molto utile.

"PRENDI" + O oppure
"PRENDI TUTTO": il giocatore può prendere tutto quello che incontra sul suo cammino.
"TIENI" + anello: l'anello di rubino e quello di bronzo hanno anche altri effetti.
"GETTA" + O: si lascia cadere un oggetto, ma bisogna stare attenti.
"MANGIA" oppure
"MANGIA" + fungo: a volte vale la pena di provare...
Quando si lotta contro un mostro, è il caso che decide il risultato del combattimento. Nei combattimenti si possono usare diverse armi, di cui il computer valuta la potenza; se l'arma non colpisce il bersaglio, torna indietro...
Buon divertimento!

Florian Bleibinhaus

```
70 REM  ****************************
80 REM  *  LA TERRA DEI 7 ANELLI  *
90 REM  *                        *
100 REM *  (C) FLORIAN BLEIBINHAUS*
110 REM *  (C) TECNICHE NUOVE     *
120 REM *      VIA MOSCOVA 46/9A  *
130 REM *                        *
140 REM ****************************
150 A=0:H=150:DIM B$(40,20):GOTO 4690
160 PRINT"■VAI VERSO "A$".":H=H-2
170 RESTORE:FOR A=1 TO (DZ-1)*17:READ A$:NEXT:IF DZ=1 THEN RESTORE
180 READ R,A$,N$,O$,S$,W$,D$,U$,DN,DO,DS,DW,DD,DU,CF,CB,CW:GOSUB1770
190 PRINT"■SEI "A$".":A=0
200 GOSUB 3410:GOTO 230
210 H=H-1
220 PRINT"■SEI "A$"."
230 IF H<1 THEN PRINT"■SEI MORTO DI FAME !":A=0:GOTO 1940
240 IF H<10 THEN PRINT"■SEI MOLTO AFFAMATO":GOTO 260
250 IF H<30 THEN PRINT"■SEI AFFAMATO"
260 E$="":INPUT E$:A=0:IF E$="L" THEN 460
270 IF E$="N" THEN IF DN<>0 THEN DZ=DN:A$="NORD":GOTO 160
280 IF E$="E" THEN IF DO<>0 THEN DZ=DO:A$="EST":GOTO 160
290 IF E$="S" THEN IF DS<>0 THEN DZ=DS:A$="SUD":GOTO 160
300 IF E$="O" THEN IF DW<>0 THEN DZ=DW:A$="OVEST":GOTO 160
310 IF E$="A" THEN IF DU<>0 THEN DZ=DU:A$="ALTO":GOTO 160
320 IF E$="B" THEN IF DD<>0 THEN DZ=DD:A$="BASSO":GOTO 160
330 IF LEFT$(E$,3)="DAI" THEN 560
340 IF LEFT$(E$,6)="PRENDI" THEN 780
350 IF LEFT$(E$,7)="DOMANDA" THEN 3180
360 IF LEFT$(E$,5)="GRIDA" THEN 2900
370 IF LEFT$(E$,5)="TIENI" THEN 2960
380 IF LEFT$(E$,5)="GETTA" THEN 3080
390 IF LEFT$(E$,6)="UCCIDI" THEN 2340
400 IF LEFT$(E$,6)="MANGIA" THEN 2050
410 IF LEFT$(E$,6)="APRITI" THEN 1840
420 IF LEFT$(E$,1)="I" THEN A=0:GOTO 1880
430 IFE$="N" OR E$="E" OR E$="S" OR E$="O" OR E$="A" OR E$="B"THEN450
440 PRINT"■UESTO NON LO POSSO TRADURRE.":GOTO 210
450 PRINT"■IMPOSSIBILE!":GOTO 210
460 REM *** 'L' (GUARDATI INTORNO) ROUTINE ***
470 IF R<>17 THEN 490
480 PRINT"■E' COSI' BUIO CHE NON VEDI NIENTE.":GOTO 210
490 PRINT"■NORD: "N$:PRINT"EST: "O$:PRINT"SUD: "S$:PRINT"OVEST: "W$
500 IF D$<>"0" THEN PRINT"BASSO: "D$
510 IF U$<>"0" THEN PRINT"ALTO :"U$
520 PRINT
```

```
530 IF B$(R,A)="" THEN GOSUB 3410:GOTO 210
540 PRINT"VEDI 1 "B$(R,A)
550 A=A+1:GOTO 530
560 REM ***ROUTINE  'DAI' ***
570 A=0:IF B$(40,0)="" THEN 710
580 IF LEN(E$)<5 THEN 710
590 PRINT:E$=RIGHT$(E$,LEN(E$)-4):IF E$="" THEN 710
600 IF E$<>"TUTTO" THEN 1270
610 PRINT"DAI 1 "B$(40,A)"."
620 B$(40,A)=""
630 IF B$(40,A+1)="" THEN 650
640 A=A+1:GOTO 610
650 PRINT"■VEDI UNA FAMIGLIA DI ORSI CHE SCAPPA    VIA.":GOTO 950
660 IF E$=B$(40,A) THEN 690
670 IF A=20 THEN 710
680 A=A+1:GOTO 660
690 PRINT"DAI 1 "E$:PRINT"■VEDI CHE UN ORSO SCAPPA VIA."
700 GOSUB 1750:GOTO 950
710 PRINT"■NON DAI NIENTE.":GOTO 950
720 B=0
730 IF B$(40,B)="" THEN RETURN
740 B=B+1:GOTO 730
750 C=0
760 IF B$(R,C)="" THEN RETURN
770 C=C+1:GOTO 760
780 REM ***ROUTINE 'PRENDI' ***
790 A=0:GOSUB 720:IF B$(R,0)="" THEN 940
800 IF LEN(E$)<8 THEN 940
810 PRINT:E$=RIGHT$(E$,LEN(E$)-7):IF E$="" THEN 940
820 IF E$<>"TUTTO" THEN 870
830 PRINT"PRENDI 1 "B$(R,A)"."
840 B$(40,B)=B$(R,A):B$(R,A)="":A=A+1
850 IF B$(R,A)="" THEN 950
860 B=B+1:GOTO 830
870 IF E$=B$(R,A) THEN 900
880 IF A=20 THEN 940
890 A=A+1:GOTO 870
900 PRINT"PRENDI 1 "E$"."
910 GOSUB 720:B$(40,B)=E$
920 B$(R,A)=B$(R,A+1):A=A+1:IF A<20 THEN 920
930 GOTO 950
940 PRINT"■NON PRENDI NIENTE.":GOTO 950
950 REM *** CONTROLLO DEI MOSTRI EECC. ***
960 IF R<>7 OR F1>0 THEN 990
970 PRINT"■IL NANO TI ATTACCA CON UN'ASCIA!":TF=2
980 MO$="IL NANO":GOSUB 1240:IF F=2 THEN 1920
990 IF R<>13 OR F3<>1 THEN 1030
1000 PRINT"■L'UOMO TI SPARA CON UNA BALESTRA !"
1010 TF=1.5:MO$="L'UOMO":GOSUB 1240
1020 IF F=2 THEN 1920
1030 IF R<>18 OR F4>0 THEN 1060
1040 PRINT"■IL FOLLETTO TI GETTA UNA LANCIA !":TF=1.3:MO$="IL FOLLETTO"
1050 GOSUB 1240:IF F=2 THEN 1920
1060 IF R<>23 OR F6=2 OR F6=3 OR F6=0 THEN 1100
1070 PRINT"■IL MAGO TI GETTA UNA PALLA DI FUOCO !"
1080 TF=1.2:MO$="IL MAGO":GOSUB 1240
1090 IF F=2 THEN 1920
1100 IF R<>25 OR F7=2 OR F7=3 OR F7=0 THEN 1140
1110 PRINT"■IL CUSTODE TI GETTA UN PUGNALE !"
```

```
1120 TF=2:MO$="IL CUSTODE":GOSUB 1240
1130 IF F=2 THEN 1920
1140 IF R=17THENPRINT"■NELL'OSCURITA' QUALCUNO TI COLPISCE":GOTO1920
1150 IF R<>33 OR F8=1 THEN 1190
1160 PRINT"■IL BRUCO TI SPUTA ADDOSSO IL SUO VELENO !"
1170 TF=2:MO$="IL BRUCO":GOSUB 1240
1180 IF F=2 THEN 1920
1190 IF R<>10 OR F0<>1 OR F2<>1 THEN 210
1200 PRINT"■IL GUARDIANO TI ATTACCA CON ARCO E FRECCE!"
1210 TF=1.4:MO$="IL GUARDIANO":GOSUB 1240
1220 IF F=2 THEN 1920
1230 GOTO 210
1240 REM *** ROUTINE PER IL RISULTATO DELLO SCONTRO***
1250 TR=INT(RND(1)*TF):IF TR=0 THEN PRINT"■"MO$" TI COLPISCE !":F=2
     :RETURN
1260 PRINT"■"MO$" NON TI COLPISCE !":F=0:RETURN
1270 REM *** SOTTOPROGRAMMA PER LA ROUTINE DAI ***
1280 GOSUB 4610:IF QW=1 THEN 1440
1290 IF R<>7 OR F1<>0 OR E$<>"FUNGO MOSCARIO"THEN 1340
1300 GOSUB 1710
1310 PRINT"■IL NANO DICE: CHE BEL FUNGO A PUNTINI   BIANCHI!"
1320 PRINT"IL NANO MANGIA IL FUNGO MOSCARIO E      MUORE."
1330 DS=8:S$="ANCORA FORESTA":F1=1:GOTO 210
1340 IF R<>10 OR F2<>1 OR E$<>"SFINGE D'ORO"OR F0=2 THEN 1380
1350 GOSUB 1710:F2=2:F0=0:DW=11
1360 PRINT"■IL GUARDIANO DICE: QUESTO CAMBIA TUTTO"
1370 PRINT"IL PONTE LEVATOIO VIENE ABBASSATO.":GOTO 210
1380 IF R<>23 OR F6>1 OR E$<>"SPADA" THEN 1410
1390 PRINT"GRAZIE! E' MOLTO RAGIONEVOLE DA PARTE   TUA.":F6=2:
     GOSUB 1710
1400 PRINT"PER QUESTO AVRAI UN ANELLO!":B$(23,0)="ANELLO DI RUBINI"
     :GOTO 210
1410 IF R<>25 OR F7>1 OR E$<>"MONETA D'ORO" THEN 1440
1420 PRINT"■IL CUSTODE DICE: ALLAH TI RICOMPENSERA'!"
1430 F7=2:GOSUB 1710:GOSUB 1770:GOTO 210
1440 A=0:IF E$="ANELLI" THEN 1470
1450 IF QW=1 THEN 710
1460 GOTO 660
1470 AB=1:E$="ANELLO":GOSUB 4610
1480 PRINT"■LA TERRA TREMA. SENTI CHE IL CASTELLO E' SCOSSO";
1490 PRINT"FINO ALLEE FONDAMENTA. TUTTE LE MURA CROLLANO."
1500 E$="ANELLO D'ORO":GOSUB 4610
1510 E$="ANELLO DI RUBINO":GOSUB 4610
1520 E$="ANELLO DI BRONZO":GOSUB 4610
1530 E$="ANELLO DI ARGENTO":GOSUB 4610
1540 E$="ANELLO DI PLATINO":GOSUB 4610
1550 E$="ANELLO DI DIAMANTE":GOSUB 4610
1560 IF AB=2 THEN PRINT"■■NON HAI TUTTI I SETTE ANELLI !":T=2:GOTO1940
1570 IF F3=2 AND F4=1 AND F8=1 THEN 1600
1580 PRINT"■HAI GLI ANELLI MA NON HAI FATTO TUTTO. ";
1590 PRINT"E' POCO, MA NON COSI' POCO PERCHE' "T=2:GOTO 1940
1600 FOR A=0 TO 999:NEXT
1610 FOR FB=0 TO 15:POKE 53280,FB:POKE 53281,15-FB:POKE 646,FB
1620 PRINT"■CE L'HAI FATTA !"
1630 FOR A=0 TO 200:NEXT:NEXT
1640 FOR A=0 TO 999:NEXT
1650 POKE 53280,10:POKE 53281,10:POKE 646,0
1660 PRINT"■■■■■   CI SEI RIUSCITO !"
```

```
1670 PRINT"■■■■HAI RISOLTO L'INCREDIBILE,";
1680 PRINT"FENOMENALE ENIGMA DEI SETTE ANELLI!"
1690 PRINT"■■■■SONO FATTI PROPRIO PER QUESTO, PER FARTI
     ARRABBIARE, CERTO !"
1700 END
1710 A=0:REM *** PERDITA DI UN OGGETTO ***
1720 IF B$(40,A)=E$ THEN 1750
1730 A=A+1:IF A<21 THEN 1720
1740 IF A=20 THEN B$(40,20)="":RETURN
1750 B$(40,A)=B$(40,A+1):A=A+1:IF A=20 THEN RETURN
1760 GOTO 1750
1770 REM *** SCOPERTA DI UN PASSAGGIO APERTO ***
1780 IF F1=1 AND R=7 THEN DS=8:S$="ANCORA FORESTA"
1790 IF F2=2 AND R=10 THEN DW=11
1800 IF F3=2 AND R=13 THEN DO=12:DN=14
1810 IF R=25 AND F7>1 THEN DU=27
1820 IF R=35 AND F9=1 THEN DW=36
1830 POKE 53281,CF:POKE 53280,CB:POKE 646,CW:RETURN
1840 REM *** ROUTINE "APRITI" ***
1850 E$="CHIAVE":GOSUB 4610:IF R<>35 THEN 440
1860 IF QW=1 THEN PRINT"■NON HAI LA CHIAVE":GOTO 210
1870 PRINT"■OK. LA PORTA E' APERTA.":F9=1:GOSUB 1770:GOTO 210
1880 REM *** INVENTARIO***
1890 PRINT"■PORTI CON TE:■":IF B$(40,0)="" THEN PRINT"NIENTE!"
1900 IF B$(40,A)="" THEN 210
1910 PRINT B$(40,A):A=A+1:GOTO 1900
1920 REM *** MORTE DEL GIOCATORE ***
1930 PRINT"■SEI MORTO !"
1940 A=0:T=T+1:IF T=3 THEN PRINT"■■■■SEI FINITO!":GOTO 1960
1950 GOTO 1980
1960 PRINT"■■■■ANCORA UNA PARTITA?":POKE 198,0:WAIT 198,1:GET A$:IF
     A$="S" THEN RUN
1970 PRINT"■":END
1980 IF B$(40,A)="" THEN 2020
1990 R=INT(RND(1)*38)+1:GOSUB 750:B$(R,C)=B$(40,A):B$(40,A)=""
2000 A=A+1:IF A=41 THEN 2020
2010 GOTO 1980
2020 PRINT"■■■SEI RINATO !":DZ=1:H=150
2030 IF F3<>2 THEN F3=0:M3=0
2040 GOTO 170
2050 REM *** ROUTINE 'MANGIA' ***
2060 IF LEN(E$)>5 THEN 2200
2070 A=0:IF R<>14 OR F5<>0 THEN 2120
2080 PRINT"■MANGIA A SAZIETA'."
2090 PRINT"BEVI TALMENTE CHE CADI SOTTO UN TAVOLO."
2100 PRINT"■LA' VEDI UNA SPADA."
2110 B$(14,0)="SPADA":F5=1:H=100:GOTO 210
2120 IF R<>15 THEN 2190
2130 PRINT"■PRENDI IL CIBO DAL FUOCO E TI SCOTTI LE DITA!"
2140 IF B$(40,0)="" THEN 210
2150 PRINT"■PER IL DOLORE LASCI CADERE 1 "B$(40,0)
2160 PRINT"UN ALBERO DI NATALE BUSSA ALLA PORTA.":B$(40,0)=""
2170 B$(40,A)=B$(40,A+1):A=A+1:IF A<20 THEN 2170
2180 GOTO 210
2190 GOTO 2330
2200 E$=RIGHT$(E$,LEN(E$)-4):GOSUB 4610:IF QW=1 THEN 2330
2210 IFE$<>"SPUGNOLO"ANDE$<>"FUNGO VELENOSO" AND E$<>"FUNGO MOSCARIO"
     THEN 2330
2220 IF B$(40,A)=E$ THEN 2240
```

```
2230 A=A+1:GOTO 2220
2240 B$(40,A)=B$(40,A+1):A=A+1:IF A<20 THEN 2240
2250 IF E$="FUNGO MOSCARIO" THEN 2300
2260 IF E$="SPUGNOLO" THEN E$=E$+" "
2270 PRINT"■1 "E$" E' UN FUNGO MANGERECCIO MOLTO GUSTOSO E ";
2280 PRINT"NUTRIENTE. TI FA BENE E TI PLACA LA FAME.":H=100
2290 GOTO 950
2300 PRINT"■MANGI IL "E$". TI FA MALE E DEVI VOMITARLO.":H=H-50
2310 IF H>1 THEN 950
2320 PRINT"■TI SEI SVUOTATO LO STOMACO. MUORI DI FAME!":GOTO 1920
2330 PRINT"■IMPOSSIBILE!":GOTO 950
2340 REM *** ROUTINE 'UCCIDI' ***
2350 IF R<>7 OR F1>0 THEN 2390
2360 TF=5:MO$="IL NANO":GOSUB 2660
2370 IF F=1 THEN F1=1:GOSUB 1770:GOTO 210
2380 GOTO 950
2390 IF R<>18 OR F4>0 THEN 2430
2400 TF=10:MO$="IL FOLLETTO":GOSUB 2660
2410 IF F=1 THEN F4=1:GOSUB 1770:GOTO 210
2420 GOTO 950
2430 IF R<>13 OR F3>1 THEN 2470
2440 F3=1:TF=3:MO$="L'UOMO":GOSUB 2660
2450 IF F<>1 THEN 950
2460 PRINT"■STAI MISERAMENTE MORENDO DI FAME.":F3=2:GOTO 1920
2470 IF R<>23 OR F6=3 THEN 2520
2480 TF=20:MO$="IL MAGO":GOSUB 2660
2490 IF F=1 THEN F6=3:GOTO 210
2500 IF F6=2 OR F6=4 THEN F6=4:GOTO 950
2510 F6=1:GOTO 950
2520 IF R<>25 OR F7=3 THEN 2570
2530 TF=2:MO$="IL CUSTODE":GOSUB 2660
2540 IF F=1 THEN F7=3:GOTO 210
2550 IF F7=2 OR F7=4 THEN F7=4:GOTO 950
2560 F7=1:GOTO 950
2570 IF R<>33 OR F8=1 THEN 2610
2580 TF=3:MO$="IL BRUCO":GOSUB 2660
2590 IF F=1 THEN F8=1:GOTO 210
2600 GOTO 950
2610 IF R<>10 OR F0=2 OR F2<>1 THEN 2650
2620 TF=15:MO$="IL GUARDIANO":GOSUB 2660
2630 F0=1:IF F=1 THEN F0=2:GOTO 210
2640 GOTO 950
2650 PRINT"■SUICIDARSI NON RISOLVE NIENTE !":GOTO 210
2660 A=0:M=.1:REM *** L'ARMA MIGLIORE PER IL GIOCATORE ***
2670 T$=B$(40,A):A=A+1
2680 IF T$="STEIN" AND M<.3 THEN M=.3
2690 IF (T$="PUGNALE" OR T$="COLTELLO") AND M<1 THEN M$=T$:M=1
2700 IF T$="LANCIA" AND M<2 THEN M=2
2710 IF T$="MAZZA DI FERRO" AND M<3 THEN M=3
2720 IF T$="ASCIA" AND M<3.5 THEN M=3.5
2730 IF T$="STELLA DEL MATTINO" AND M<4 THEN M=4
2740 IF T$="SPADA" AND M<5 THEN M=5
2750 IF A<21 THEN 2670
2760 IF M=.1 THEN PRINT"■PROVI UN COLPO DI KARATE'!"
2770 IF M=.3 THEN PRINT"■GETTA UNA PIETRA !"
2780 IF M=1 THEN PRINT"■ATTACCHI SELVAGGIAMENTE CON IL "M$"!"
2790 IF M<>3 THEN 2810
2800 PRINT"■COLPISCI CON LA MAZZA IL TUO AVVERSARIO !"
2810 IF M=5 THEN PRINT"■LO ATTACCHI CON LA SPADA FATATA!"
```

```
2820 IF M=2 THEN PRINT"■TI GETTI SUL TUO NEMICO CON LA LANCIA!"
2830 IF M<>4 THEN 2850
2840 PRINT"■COLPISCI IL TUO AVVERSARIO CON UN FENDENTE !"
2850 IF M=3.5 THEN PRINT "■LO COLPISCI, E' MORTO!"
2860 TF=TF/M
2870 TR=INT(RND(1)*TF)
2880 IF TR=0 THEN PRINT"■LO COLPISCI!"MO$" E' MORTO!":F=1:RETURN
2890 PRINT"■NON LO COLPISCI !":F=0:RETURN
2900 REM *** ROUTINE 'GRIDA' ***
2910 PRINT"■LANCI UN GRIDO":IF R<>22 THEN 2950
2920 PRINT"■LE VECCHIE MURA CROLLANO."
2930 PRINT"RESTI SEPOLTO SOTTO LE MACERIE."
2940 PRINT"RIMARRAI LA' PER L'ETERNITA'!":GOTO 1960
2950 PRINT"■NON SUCCEDE NIENTE.":GOTO 950
2960 REM *** ROUTINE 'TIENI' ***
2970 E$=RIGHT$(E$,LEN(E$)-5)
2980 GOSUB 4610:IF QW=1 THEN 3070
2990 IF E$<>"ANELLO DI RUBINO" OR R<>17 THEN 3030
3000 PRINTE$ "■EMANA ABBASTANZA LUCE."
3010 PRINT"■NORD: "N$:PRINT"EST: "O$:PRINT"SUD: "S$:PRINT"OVEST: "W$
3020 PRINT"BASSO :"D$:PRINT"ALTO :"U$:DD=39:GOTO 210
3030 IF R<>39 OR E$<>"ANELLO DI BRONZO" THEN 3070
3040 PRINT"■■L'ANELLO DI BRONZO IMMERGE LA STANZA IN
     UNA LUCE DORATA. SUL";
3050 PRINT"TETRAEDO APPARE UNA SCRITTA: METTI I SETTE
     ANELLI SULL'ALTARE!"
3060 PRINT"UNA LEGGERA SCOSSA FA TREMARE IL PAVIMENTO !":FF=1:GOTO 210
3070 PRINT"■NON SUCCEDE NIENTE":GOTO 950
3080 REM *** ROUTINE 'GETTA' ***
3090 IF LEN(E$)<6 THEN 3150
3100 E$=RIGHT$(E$,LEN(E$)-5):GOSUB 4610:IF QW=1 THEN 3150
3110 IF R<>10 OR E$<>"PIETRA" THEN 3160
3120 F2=1:A=0
3130 PRINT"■GETTI UNA PIETRA CONTRO IL PONTE LEVATOIO.
     ARRIVA UN GUARDIANO."
3140 GOSUB 1710:GOTO 210
3150 PRINT"■IMPOSSIBILE!":GOTO 950
3160 PRINT"■GETTI 1 "E$".":PRINT"NON SUCCEDE NIENTE.":GOSUB 750:
     GOSUB 1710
3170 B$(R,C)=E$:GOTO 950
3180 REM *** ROUTINE 'DOMANDA' ***
3190 IF R<>7 OR F1<>0 THEN 3220
3200 PRINT"■IL NANO DICE: SCIAGURATO! OSI PENETRARE NEL MIO REGNO?"
3210 PRINT"QUESTO TI COSTERA' LA VITA!!!":GOTO 950
3220 IF R<>10 OR F2<>1 THEN 3250
3230 PRINT"IL GUARDIANO DICE: DI QUI NON ENTRERAI."
3240 GOTO 950
3250 IF R<>13 OR F3<>0 THEN 3310
3260 PRINT"L'UOMO DICE: COS'E' CHE "
3270 PRINT"PIU' SI TIRA PIU' SI ACCORCIA?"
3280 INPUT E$:IF LEFT$(E$,5)<>"IL SIGARO" THEN 3300
3290 DN=14:DO=12:PRINT"■L'UOMO DICE: GIUSTO!":F3=2:GOTO 210
3300 PRINT"■L'UOMO DICE: SBAGLIATO !":F3=1:GOTO 950
3310 IF R=22 THEN PRINT"■TU MORMORI QUALCOSA, MOLTO PIANO.":GOTO 210
3320 IF R<>23 OR F6=>3 THEN 3350
3330 PRINT"IL MAGO DICE: MI HAI PORTATO LA SPADA"
3340 PRINT"MAGICA?":GOTO 950
3350 IF R<>25 OR F7>0 THEN 3370
3360 PRINT"■IL CUSTODE DICE: HAI UNA MONETINA?"
```

```
3370 IF R<>18 OR F4<>0 THEN 3400
3380 PRINT"■IL FOLLETO DICE: HUMBA TSCHUGA TUGEDUG!"
3390 PRINT"E SI BATTE IL PETTO.":GOTO 950
3400 PRINT "■TI CHIEDI COSA SIGNIFICHI TUTTO CIO'!":GOTO 950
3410 REM *** OUTPUT DELLE INFORMAZIONI SU UN LUOGO ***
3420 IF R=7 AND F1=0 THEN PRINT"■VEDI UN NANO CHE TI SALTELLA INTORNO."
3430 IF R=10 AND F2<>2 THEN PRINT"■IL PONTE LEVATOIO E' ALZATO."
3440 IF R<>12 THEN 3470
3450 PRINT"■NONOSTANTE IL BEL TEMPO QUI' E' FREDDO E BUIO."
3460 IF R=24 THEN PRINT"■QUI C'E' UN ODORE PESTILENZIALE!"
3470 IF R<>37 OR R<>38 THEN 3490
3480 PRINT"■LE STALATTITI TI IMPEDISCONOO DI VEDERE."
3490 IF R<>13 OR F3<>0 OR M3=1 THEN 3530
3500 PRINT"■MOLTO LENTAMENTE TI ABITUI ALL'OSCURITA'."
3510 PRINT"VEDI UN UOMO SEDUTO SU UNA SEDIA."
3520 PRINT"SENTI CHE DIETRO DI TE SI CHIUDE UNA PORTA.":M3=1
3530 IFR=14ANDF5=0THENPRINT"■LA TAVOLA E' SONTUOSAMENTE IMBANDITA."
3540 IF R<>15 THEN 3560
3550 PRINT"■VEDI UN FOCOLARE.":PRINT"SUL FUOCO CUOCE DEL CIBO."
3560 IF R<>18 OR F4<>0 THEN 3580
3570 PRINT"■UN FOLLETTO SI SCAGLIA CONTRO DI TE URLANDO !"
3580 IF R<>23 THEN 3610
3590 IFF6=0ORF6=1THEN PRINT"■IL MAGO E' DI FRONTE A TE MINACCIOSO."
3600 IF F6=2 THEN PRINT"■IL MAGO TI GUARDA AMICHEVOLMENTE."
3610 IF R=25 AND F7<>3 THEN PRINT"■VEDI UN GUARDIANO."
3620 IF R<>31 THEN 3700
3630 PRINT"■HAI UNA BELLISSIMA VISTA."
3640 PRINT"SOTTO DI TE VEDI IL CASTELLO ABBANDONATO."
3650 PRINT"DIETRO LE MURA DEL CASTELLO C'E' UN GIARDINO CHE";
3660 PRINT"POI DIVENTA UN FITTO BOSCO."
3670 PRINT"IN FONDO TORREGGIANO LE CIME DELLE MONTAGNE."
3680 PRINT"UN FIUME SCORRE SERPEGGIANDO VERSO SUD."
3690 PRINT"VERSO NORD E VERSO EST UNA FITTA NEBBIA IMPEDISCE
     LA VISTA."
3700 IF R<>39 THEN 3740
3710 PRINT"■DAL TERRENO VIENE EMANATA UNA LUCE ROSSASTRA.
     DAVANTI A TE ";
3720 PRINT"C'E' UN TETRAEDRO ALTO COME UN UOMO. ALLE PARETI
     CI SONO DEGLI";
3730 PRINT" ORNAMENTI CHE RAPPRESENTANO DELLE DIVINITA' GRECHE."
3740 IF R=33 AND F8=0 THEN PRINT"■VEDI UN GIGANTESCO BRUCO!"
3750 IF R=35 THEN PRINT"■VEDI UNA PORTA SEGRETA."
3760 IF R=30 OR R=26 THEN PRINT"■UNA LUCE DEBOLE FILTRA DALL'ALTO."
3770 IF R=20 THEN PRINT"■UNA LUCE DEBOLISSIMA FILTRA DALL'ALTO."
3780 RETURN
3790 DATA 1,IN UNA FORESTA DI QUERCE,ANCORA FORESTA,ANCORA FORESTA
     ,ANCORA FORESTA
3800 DATA UN SENTIERO,0,0,2,6,9,10,0,0,9,9,5
3810 DATA2,IN UNA FORESTA DI QUERCE,ANCORA FORESTA,UNA FITTA
     FORESTA,ANCORA FORESTA
3820 DATA UN FIUME,0,0,3,0,1,0,0,0,9,9,5
3830 DATA 3,IN UN BOSCO MISTO,UN FIUME,ANCORA FORESTA,ANCORA FORESTA
3840 DATA UN FIUME,0,0,0,4,2,0,0,0,9,9,10
3850 DATA 4,IN UN BOSCO DI ABETI,UN FIUME,UNA PARETE DI PIETRA
     ,ANCORA FORESTA
3860 DATA ANCORA FORESTA,0,0,0,0,5,3,0,0,9,9,14
3870 DATA 5,IN UNA RADURA,ANCORA FORESTA,UNA PARETE DI PIETRA
     ,ANCORA FORESTA
3880 DATA UNA FORESTA MOLTO FITTA,UN BUCO,0,4,0,6,0,32,0,6,6,0
```

```
3890 DATA6,IN UN BOSCO DI QUERCE,ANCORA FORESTA,UNA PARETE DI
     PIETRA,ANCORA FORESTA
3900 DATA ANCORA FORESTA,0,0,5,0,7,1,0,0,9,9,5
3910 DATA 7,IN UN BOSCO INCANTATO,ANCORA FORESTA,UN BOSCO MOLTO FITTO
3920 DATA UN BOSCO MOLTO FITTO,ANCORA FORESTA,0,0,6,0,0,9,0,0,9,9,5
3930 DATA 8,IN UN BOSCO INCANTATO,ANCORA FORESTA,UNA PARETE DI PIETRA
3940 DATA UN BOSCO MOLTO FITTO,UN FIUME,0,0,7,0,0,0,0,0,9,9,5
3950 DATA 9,IN UN BOSCO DI QUERCE,ANCORA FORESTA,ANCORA FORESTA
     ,UN FIUME
3960 DATA UN FIUME,0,0,1,7,0,0,0,0,9,9,5
3970 DATA 10,DAVANTI AL CASTELLO,UN FIUME,UN SENTIERO,UN FIUME
3980 DATA UN PONTE LEVATOIO,0,0,0,1,0,0,0,0,6,9,5
3990 DATA 11,UN PONTE LEVATOIO,IL FOSSATO,UN SENTIERO,IL FOSSATO
4000 DATA UN PORTALE,0,0,0,10,0,12,0,0,6,11,5
4010 DATA 12,NEL CORTILE,UNA PORTA,UN PORTALE,UN MURO,UNA PORTA,0
4020 DATA 0,15,11,0,13,0,0,6,11,0
4030 DATA 13,IN UN LUOGO BUIO,UNA PORTA,UNA PORTA,UNA PARETE
4040 DATA UNA PARETE,0,0,0,0,0,0,0,0,11,0,0
4050 DATA 14,IN SALA DA PRANZO,UNA PARETE,UNA PORTA,UNA PORTA
     ,UNA PARETE
4060 DATA 0,0,0,15,13,0,0,0,11,11,14
4070 DATA 15,NELLA CUCINA,UNA PORTA,UNA PARETE,UNA PORTA,UNA PORTA,0,0
4080 DATA 16,0,12,14,0,0,11,11,7
4090 DATA 16,NELLA CAMERA DA LETTO,UNA PARETE,UNA PARETE,UNA
     PORTA,UNA PARETE
4100 DATA 0,UNA SCALA,0,0,15,0,0,18,11,14,14
4110 DATA 17,NELL'OSCURITA' PIU' COMPLETA,UNA PARETE,UNA PARETE
4120 DATA UNA PARETE,UNA PARETE,OSCURITA',OSCURITA',0,0,0,0,0,20,0,0,1
4130 DATA 18,PRIMA DELLO SCALONE,UN CORRIDOIO,UNA PARETE,UNA PARETE
4140 DATA UNA PARETE,UNA SCALA,0,19,0,0,0,16,0,11,11,0
4150 DATA 19,SULLE SCALE,UNA PARETE,UNA PARETE,UN CORRIDOIO
     ,UNA PARETE,0
4160 DATA UNA SCALA,0,0,18,0,0,21,11,11,14
4170 DATA 20,NELLA TORRE,UNA PARETE,UNA PARETE,UN CORRIDOIO,UNA PARETE
4180 DATA UNA SCALA DI PIETRA,UNA SCALA DI PIETRA,0,0,0,0,17,26
     ,11,11,0
4190 DATA 21,SULLA SCALA,UNA PARETE,UNA PARETE,UNA PARETE,UNA PORTA
4200 DATA UNA SCALA,0,0,0,0,22,19,0,11,11,14
4210 DATA 22,NEL SILENZIO PIU' ASSOLUTO,UNA PARETE,UNA PORTA
     ,UNA PARETE
4220 DATA UNA PORTA DI LEGNO,0,0,0,21,0,23,0,0,11,11,14
4230 DATA 23,NEL REGNO DEL GRANDE MAGO DEL CASTELLO DI CANTERVILLE
4240 DATA UNA PORTA,UNA PORTA DI LEGNO, UNA PARETE
4250 DATA UN BELLISSIMO GOBELIN,0,0,24,22,0,0,0,0,6,2,0
4260 DATA 24,NELLA CASA DEL GRANDE MAGO DI CANTERVILLE,UNA PARETE
4270 DATA UNA PORTA,UNA PORTA,UNA PARETE,0,0,0,25,23,0,0,0
4280 DATA 6,2,0
4290 DATA 25,IN UNA GALLERIA,UN QUADRO DEL CAVALIERE BALDUIN
     ,ANCORA UN QUADRO
4300 DATA UN ALTRO QUADRO,UNA PORTA,0,UNA SCALA,0,0,0,24,0,0,2,2,0
4310 DATA 26,NELLA TORRE,UNA PARETE,UNA PARETE,UNA PARETE,UNA PARETE
4320 DATA UNA SCALA DI PIETRA,UNA SCALA DI PIETRA,0,0,0,0,20,29
     ,11,14,0
4330 DATA 27,UNA SALA D'ARMI,UN ARCO,UNA ARMATURA DEL RE ARTU'
4340 DATA UN'ALTRA ARMATURA,ANCORA UN'ARMATURA,UNA SCALA,0
4350 DATA 28,0,0,0,25,0,11,11,0
4360 DATA 28,NELL'ATRIO DELLA TORRE,UNA PARETE,UNA PARETE,UN ARCO
4370 DATA UNA PORTA,0,0,0,0,27,29,0,0,11,11,14
4380 DATA 29,NELLA TORRE,UNA PARETE,UNA PORTA,UNA PARETE,UNA PARETE
```

```
4390 DATA UNA SCALA DI PIETRA,UNA SCALA DI PIETRA,0,28,0,0,26,30
     ,11,14,14
4400 DATA 30,NELLA TORRE,UNA PARETE,UNA PARETE,UNA PARETE,UNA PARETE
4410 DATA UNA SCALA DI PIETRA,UNA SCALA DI PIETRA,0,0,0,0,29,31
     ,11,14,14
4420 DATA 31,SULLA CIMA DELLA TORRE,MERLI,MERLI,MERLI,MERLI
4430 DATA UNA SCALA DI PIETRA,0,0,0,0,0,30,0,6,6,14
4440 DATA 32,IN UNA CAVERNA,UN CORRIDOIO,STALATTITI,UNA PARETE
4450 DATA STALATTITI,0,UNA GROSSA APERTURA,33,0,0,0,0,5,11,0,14
4460 DATA 33,IN UN PASSAGGIO SOTTERRANEO,UNA PARETE,STALATTITI
4470 DATA UN CORRIDOIO,UN CORRIDOIO,0,0,0,0,32,34,0,0,11,0,0
4480 DATA 34,IN UNA GROTTA,UNA PARETE,UN CORRIDOIO,UN BUCO,STALATTITI
4490 DATA 0,0,0,33,35,0,0,0,11,0,0
4500 DATA 35,IN UN GROTTA SEGRETA,UNA PARETE,UNA PORTA,UNA PARETE
     ,UNA PARETE
4510 DATA UNA STANZA SEGRETA,0,0,34,0,37,0,0,0,11,11,0
4520 DATA 36,NELLA STANZA SEGRETA,UNA PARETE,UNA PORTA,UNA PARETE
     ,UNA PARETE,0
4530 DATA 0,0,35,0,0,0,0,11,0,0
4540 DATA 37,UN PASSAGGIO SOTTERRANEO,UN BUCO,STALATTITI
4550 DATA UNA PARETE,UNA PARETE,0,0,35,38,0,0,0,0,11,11,0
4560 DATA 38,IN UNA CAVERNA,UNA PARETE,UNA PARETE,UNA PARETE
4570 DATA STALATTITI,0,0,0,0,0,37,0,0,11,11,0
4580 DATA 39,NEL CARCERE DEL CASTELLO DI CANTERVILLE NELLA
     TERRA DEI SETTE ANELLI
4590 DATA UNA PARETE,UNA PARETE,UNA PARETE,UNA PARETE
4600 DATA UN PAVIMENTO FOSFORESCENTE,OSCURITA',0,0,0,0,0,17,0,0,2
4610 REM *** IL GIOCATORE HA DEGLI OGGETTI ? ***
4620 A=0:QW=0
4630 IF B$(40,A)=E$ THEN 4660
4640 A=A+1:IF A=21 THEN QW=1:GOTO 4660
4650 GOTO 4630
4660 IF AB=0 THEN RETURN
4670 IF QW=1 THEN AB=2
4680 RETURN
4690 REM *** PREPARAZIONE DEL GIOCO ***
4700 REM *** ASSEGNAZIONE DEGLI OGGETTI ***
4710 B$(2,0)="FUNGO MOSCARIO":B$(3,0)="MONETA D'ORO":B$(4,0)="PUGNALE"
4720 B$(6,0)="SFINGE D'ORO":B$(7,0)="SPUGNOLO":B$(8,0)=
     "ANELLO DI DIAMANTE"
4730 B$(9,0)="FUNGO VELENOSO":B$(10,0)="PIETRA":B$(18,0)=
     "STELLA DEL MATTINO"
4740 B$(19,0)="ANELLO D'ORO":B$(21,0)="CHIAVE"
4750 B$(22,0)="COLTELLO":B$(27,0)="LANCIA":B$(27,1)="MAZZA"
4760 B$(31,0)="ANELLO D'ARGENTO":B$(34,0)="ASCIA":B$(36,0)="ANELLO"
4770 B$(39,0)="ANELLO DI PLATINO":B$(38,0)="ANELLO DI BRONZO"
4780 PRINT CHR$(8) CHR$(142):POKE 53280,10:POKE 53281,10:POKE 646,0
4790 PRINT"■F.B. PROD. PRESENTA :",,,
     "■■■■■■■■■■■■ LA TERRA DEI SETTE ANELLI■■■■■■■■■"
4800 GOSUB 4930
4810 PRINT"■■■ABBREVIAZIONE DEI COMANDI IMPORTANTI :"
4820 PRINT"■■■■■L■ = GUARDATI INTORNO    ■I■ = INVENTARIO"
4830 PRINT"■■N■ = VAI A NORD       "
4840 PRINT"■■S■ = VAI A SUD            ■A■ = VAI IN ALTO"
4850 PRINT"■■O■ = VAI A OVEST      "
4860 PRINT"■■E■ = VAI A EST            ■B■ = VAI IN BASSO"
4870 GOSUB 4930
4880 PRINT"■■■■■GLI ARTICOLI SI DEVONO TRALASCIARE."
4890 PRINT"■■■■■E' CONSIGLIABILE USARE 'TUTTO' CON      'PRENDI' ";
```

```
4910 GOSUB 4930:PRINT"■■■■COMANDI:          ",,,"■DAI","PRENDI","APRI"
     ,"GRIDA"
4920 PRINT"■TIENI","CHIEDI","UCCIDI","GETTA","■MANGIA":GOSUB 4930
     :GOTO 180
4930 PRINT,,"■■<RETURN>":POKE 198,0:WAIT 198,1:GET A$
4940 PRINT"■■","■■A TERRA DEI SETTE ANELLI":RETURN
```

Le coniugazioni latine

Lunghezza del programma: circa 7500 byte. Linguaggio di programmazione: BASIC. Accessori: registratore.

EBO, EBIS, EBIT, EBIMUS, EBITIS, EBUNT... la coniugazione dei verbi latini si impara meglio con l'aiuto del computer. Il programma è suddiviso in una parte di esercizio, una di analisi e una di congiunzione.

ESERCIZIO

Nelle linee da 1810 a 1990 il computer genera una forma verbale, determinata a caso (alle linee 1770-1800). Per ogni coniugazione vengono dati tre verbi (le DATA da 1660 a 1700), per un totale di 15 verbi. L'utente deve essere in grado di riconoscere queste forme e ne deve indicare esattamente la coniugazione, il tempo, il modo e infine la persona. Gli input dell'utente vengono controllati ed eventualmente corretti. In seguito il computer aspetta che venga battuto un tasto, S per sì o N per no, per sapere se deve continuare con un'altra forma verbale.

ANALISI

È il processo contrario a quello usato precedentemente. Nelle linee da 200 a 450 è l'utente a suggerire una forma verbale e il computer la analizza. Per "radice" di un verbo si intende la forma dell'infinito senza desinenza, per esempio: laudare - laud.

CONIUGAZIONE

Dopo l'introduzione dell'infinito e del tempo desiderato, il computer coniuga il verbo alle linee 600 - 1000.
— Presente
— Imperfetto
— Futuro
La forma può essere attiva o passiva, inoltre si può scegliere il modo indicativo o il modo congiuntivo (in questo caso il futuro non esiste). Perciò si hanno dieci possibili combinazioni per ogni coniugazione.

ELENCO DELLE VARIABILI

E$ () - desinenze, dove E$ (X,Y,Z) significa: X - coniugazione, Y - tempo e modo, Z - persona (1 - 6)
F0$ - forma del verbo
ST$ - radice
S$ - desinenza (separata dalla radice del verbo)
VE$ - infinito del verbo
KO$ = str$(KO) - numero della coniugazione
Z$ = str$ (Z) - numero del tempo e modo
R$ - forma attiva (F5)/forma passiva (F7)
W$ - radice del verbo
VB$ () - verbo esemplificativo, dove VB$ (X,Y) significa: X - numero della coniugazione, Y - numero del verbo (0-2)

X - generatore casuale: numero della coniugazione
Y - generatore casuale: numero del tempo e modo
Z - generatore casuale: numero della persona
KJ = val (KJ$) - numero della coniugazione
FM = val (FM$) - forma verbale, cioè tempo e modo
PR = val (PR$) - persona (1-3 singolare, 4-6 plurale)
Variabili di aiuto: I, J, H, R$, R.

Thomas Zurek

```
100 REM **************************
110 REM ***** CONIUGAZIONI ******
120 REM **************************
130 REM ** (C) THOMAS ZUREK     **
135 REM ** (C) TECNICHE  NUOVE **
140 REM **************************
150 POKE53280,0:POKE53281,0
160 CLR:RESTORE
170 FORI=1TO5:REM CONIUGAZIONE
180 FORJ=1TO10:REM TEMPO + ATT./PASS.
190 FORH=1TO6:REM PERSONA
200 READE$(I,J,H)
210 NEXTH,J,I
220 PRINTCHR$(142);
230 PRINTCHR$(147);CHR$(5);"**     V E R B I   L A T I N I        **"
235 PRINT
240 PRINTCHR$(154);"          1  ESERCIZIO":PRINT
250 PRINTCHR$(156);"          2  ANALISI":PRINT
260 PRINTCHR$(154);"          3  DECLINAZIONE"
270 GETR$:IFR$=""THEN270
280 R=ASC(R$):IFR<49ORR>51THEN270
290 R=VAL(R$):ONRGOTO1710,300,680
300 REM *** ANALISI ***
310 PRINTCHR$(147);"INTRODUCETE LA FORMA VERBALE"
320 PRINT"E LA CONIUGAZIONE :"
330 PRINT:INPUT"FORMA";FO$:PRINT
340 INPUT"RADICE";ST$:PRINT
350 PRINTCHR$(159);"       1  CONIUGAZIONE IN  A"
360 PRINT:PRINTCHR$(158);"        2  CONIUGAZIONE IN  E"
370 PRINT:PRINTCHR$(159);"        3  CONIUGAZIONE IN  I"
380 PRINT:PRINTCHR$(158);"        4  CONIUG. CONSONATICA"
390 PRINT:PRINTCHR$(159);"        5  CONIUGAZIONE MISTA"
400 GETR$:IFR$=""THEN400
410 IFASC(R$)<49ORASC(R$)>53THEN400
420 R=VAL(R$):PRINT:PRINT
430 S$=RIGHT$(FO$,LEN(FO$)-LEN(ST$))
440 FORJ=1TO10
450 FORH=1TO6
460 IFE$(R,J,H)=S$THENGOSUB480
470 NEXTH,J:GOTO630
480 PRINTCHR$(154);:ONJGOTO490,500,510,520,530,540,550,560,570,580
490 PRINT"PRESENTE INDICATIVO ATTIVO":GOTO590
500 PRINT"IMPERFETTO INDICATIVO ATTIVO":GOTO590
510 PRINT"FUTURO INDICATIVO ATTIVO":GOTO590
520 PRINT"PRESENTE INDICATIVO PASSIVO":GOTO590
530 PRINT"IMPERFETTO INDICATIVO PASSIVO":GOTO590
540 PRINT"FUTURO INDICATIVO PASSIVO":GOTO590
550 PRINT"PRESENTE CONGIUNTIVO ATTIVO":GOTO590
560 PRINT"IMPERFETTO CONGIUNTIVO ATTIVO":GOTO590
```

```
570 PRINT"PRESENTE CONGIUNTIVO PASSIVO":GOTO590
580 PRINT"IMPERFETTO CONGIUNTIVO PASSIVO"
590 PRINTCHR$(159):IFH>3ANDH<7THENPRINTH-3".PERSONA PLURALE"
600 IFH<4THENPRINTHCHR$(157);".PERSONA SINGOLARE"
610 IFH>6THENPRINTCHR$(145);CHR$(145);"QUESTA FORMA NON ESISTE !"
620 PRINT:RETURN
630 PRINT:PRINTCHR$(28);"ANCORA UNA VOLTA (F5/F7) ?";CHR$(129)
640 GETR$:IFR$=""THEN640
650 IFR$=CHR$(135)THEN160
660 IFR$<>CHR$(136)THEN640
670 END
680 REM *** DECLINAZIONE ***
690 PRINTCHR$(147);CHR$(5)"**  D E C L I N A Z I O N E **":
    PRINT:PRINTCHR$(156)
700 PRINT"QUESTO PROGRAMMA, DOPO L'INPUT DI UN":PRINT
710 PRINT"VERBO LO DECLINA NEL TEMPO E NEL MODO   DESIDERATO.":
    PRINT:PRINTCHR$(129)
720 INPUT"INFINITO ";VE$
730 S$=RIGHT$(VE$,1)
740 IFS$<>"E"ANDS$<>"I"THEN720
750 PRINT
760 PRINTCHR$(154);"          1  CONIUGAZIONE IN A":PRINT
770 PRINTCHR$(152);"          2  CONIUGAZIONE IN E":PRINT
780 PRINTCHR$(154);"          3  CONIUGAZIONE IN I":PRINT
790 PRINTCHR$(152);"          4  CONIUG. CONSONATICA":PRINT
800 PRINTCHR$(154);"          5  CONIUGAZIONE MISTA"
810 GETKO$:IFKO$=""THEN810
820 IFASC(KO$)<49ORASC(KO$)>53THEN810
830 KO=VAL(KO$)
840 PRINTCHR$(147);
850 PRINTCHR$(151);"        1  PRESENTE INDICATIVO":PRINT
860 PRINTCHR$(155);"        2  IMPERFETTO INDICATIVO":PRINT
870 PRINTCHR$(151);"        3  FUTURO INDICATIVO":PRINT
880 PRINTCHR$(155);"        4  PRESENTE CONGIUNTIVO":PRINT
890 PRINTCHR$(151);"        5  IMPERFETTO CONGIUNTIVO":PRINT
900 GETZ$:IFZ$=""THEN900
910 IFASC(Z$)<49ORASC(Z$)>53THEN900
920 Z=VAL(Z$)
930 IFS$="I"THEN970
940 PRINTCHR$(154);"ATTIVO (F5) / PASSIVO (F7) ?"
950 GETR$:IFR$=""THEN950
960 IFR$<>CHR$(135)ANDR$<>CHR$(136)THEN950
970 PRINTCHR$(147);CHR$(158);VE$;
980 IFS$="I"ANDZ<=3THENZ=Z+3:R$=CHR$(136):GOTO1030
990 IFS$="I"ANDZ>3THENZ=Z+3:R$=CHR$(136):GOTO1020
1000 IFZ<4ANDR$=CHR$(136)THENZ=Z+3:GOTO1030
1010 IFZ>3THENZ=Z+3
1020 IFZ>3ANDR$=CHR$(136)THENZ=Z+2
1030 IFS$="I"ANDKO>3THENW$=LEFT$(VE$,LEN(VE$)-1)
1040 IFS$="I"ANDKO<4THENW$=LEFT$(VE$,LEN(VE$)-3)
1050 IFS$="E"THENW$=LEFT$(VE$,LEN(VE$)-3)
1060 FORH=1TO6
1070 PRINTTAB(14);CHR$(159);W$+E$(KO,Z,H)
1080 NEXTH
1090 GOTO630
1100 REM *** CONIUGAZIONE IN A ***
1110 DATA"O","AS","AT","AMUS","ATIS","ANT"
1120 DATA"ABAM","ABAS","ABAT","ABAMUS","ABATIS","ABANT"
1130 DATA"ABO","ABIS","ABIT","ABIMUS","ABITIS","ABUNT"
```

```
1140 DATA"OR","ARIS","ATUR","AMUR","AMINI","ANTUR"
1150 DATA"ABAR","ABARIS","ABATUR","ABAMUR","ABAMINI","ABANTUR"
1160 DATA"ABOR","ABERIS","ABITUR","ABIMUR","ABIMINI","ABUNTUR"
1170 DATA"EM","ES","ET","EMUS","ETIS","ENT"
1180 DATA"AREM","ARES","ARET","AREMUS","ARETIS","ARENT"
1190 DATA"ER","ERIS","ETUR","EMUR","EMINI","ENTUR"
1200 DATA"ARER","ARERIS","ARETUR","AREMUR","AREMINI","ARENTUR"
1210 REM *** CONIUGAZIONE IN E ***
1220 DATA"EO","ES","ET","EMUS","ETIS","ENT"
1230 DATA"EBAM","EBAS","EBAT","EBAMUS","EBATIS","EBANT"
1240 DATA"EBO","EBIS","EBIT","EBIMUS","EBITIS","EBUNT"
1250 DATA"EOR","ERIS","ETUR","EMUR","EMINI","ENTUR"
1260 DATA"EBAR","EBARIS","EBATUR","EBAMUR","EBAMINI","EBANTUR"
1270 DATA"EBOR","EBERIS","EBITUR","EBIMUR","EBIMINI","EBUNTUR"
1280 DATA"EAM","EAS","EAT","EAMUS","EATIS","EANT"
1290 DATA"EREM","ERES","ERET","EREMUS","ERETIS","ERENT"
1300 DATA"EAR","EARIS","EATUR","EAMUR","EAMINI","EANTUR"
1310 DATA"ERER","ERERIS","ERETUR","EREMUR","EREMINI","ERENTUR"
1320 REM *** CONIUGAZIONE IN I ***
1330 DATA"IO","IS","IT","IMUS","ITIS","IUNT"
1340 DATA"IEBAM","IEBAS","IEBAT","IEBAMUS","IEBATIS","IEBANT"
1350 DATA"IAM","IES","IET","IEMUS","IETIS","IENT"
1360 DATA"IOR","IRIS","ITUR","IMUR","IMINI","IUNTUR"
1370 DATA"IEBAR","IEBARIS","IEBATUR","IEBAMUR","IEBAMINI","IEBANTUR"
1380 DATA"IAR","IERIS","IETUR","IEMUR","IEMINI","IENTUR"
1390 DATA"IAM","IAS","IAT","IAMUS","IATIS","IANT"
1400 DATA"IREM","IRES","IRET","IREMUS","IRETIS","IRENT"
1410 DATA"IAR","IARIS","IATUR","IAMUR","IAMINI","IANTUR"
1420 DATA"IRER","IRERIS","IRETUR","IREMUR","IREMINI","IRENTUR"
1430 REM *** CONIUG. CONSONATICA ***
1440 DATA"O","IS","IT","IMUS","ITIS","UNT"
1450 DATA"EBAM","EBAS","EBAT","EBAMUS","EBATIS","EBANT"
1460 DATA"AM","ES","ET","EMUS","ETIS","ENT"
1470 DATA"OR","ERIS","ITUR","IMUR","IMINI","UNTUR"
1480 DATA"EBAR","EBARIS","EBATUR","EBAMUR","EBAMINI","EBANTUR"
1490 DATA"AR","ERIS","ETUR","EMUR","EMINI","ENTUR"
1500 DATA"AM","AS","AT","AMUS","ATIS","ANT"
1510 DATA"EREM","ERES","ERET","EREMUS","ERETIS","ERENT"
1520 DATA"AR","ARIS","ATUR","AMUR","AMINI","ANTUR"
1530 DATA"ERER","ERERIS","ERETUR","EREMUR","EREMINI","ERENTUR"
1540 REM *** CONIUGAZIONE MISTA ***
1550 DATA"IO","IS","IT","IMUS","ITIS","IUNT"
1560 DATA"IEBAM","IEBAS","IEBAT","IEBAMUS","IEBATIS","IEBANT"
1570 DATA"IAM","IES","IET","IEMUS","IETIS","IENT"
1580 DATA"IOR","ERIS","ITUR","IMUR","IMINI","IUNTUR"
1590 DATA"IEBAR","IEBARIS","IEBATUR","IEBAMUR","IEBAMINI","IEBANTUR"
1600 DATA"IAR","IERIS","IETUR","IEMUR","IEMINI","IENTUR"
1610 DATA"IAM","IAS","IAT","IAMUS","IATIS","IANT"
1620 DATA"EREM","ERES","ERET","EREMUS","ERETIS","ERENT"
1630 DATA"IAR","IARIS","IATUR","IAMUR","IAMINI","IANTUR"
1640 DATA"ERER","ERERIS","ERETUR","EREMUR","EREMINI","ERENTUR"
1650 REM *** VERBI ESEMPLIFICATIVI (RADICE) ***
1660 DATA"LABOR","LAUD","IMPER"
1670 DATA"MON","DEB","FL"
1680 DATA"AUD","IMPED","SAEV"
1690 DATA"REG","DIC","PON"
1700 DATA"RAP","CAP","PAR"
1710 REM ***** ESERCIZIO *****
1720 FORI=1TO5
```

```
1730 FORJ=0TO2
1740 READVB$(I,J)
1750 NEXTJ,I
1760 PRINTCHR$(147);CHR$(5);"U E B E N  :  ";
1770 X=INT(RND(1)*5+1):REM CONIUGAZIONE
1780 Y=INT(RND(1)*3):REM VERBI ESEMPI
1790 Z=INT(RND(1)*10+1):REM FORMA
1800 P=INT(RND(1)*6+1):REM PERSONA
1810 PRINTCHR$(159);VB$(X,Y)+E$(X,Z,P):PRINT
1820 PRINTCHR$(154);"CONIUGAZIONE:  1  CONOIGAZIONE IN A"
1830 PRINTCHR$(156);"               2  CONIUGAZIONE IN E"
1840 PRINTCHR$(154);"               3  CONIUGAZIONE IN I"
1850 PRINTCHR$(156);"               4  CONIUG. CONSONATICA"
1860 PRINTCHR$(154);"               5  CONIUGAZIONE MISTA"
1870 PRINT:PRINTCHR$(156);"FORMA       :  0  PRESENTE ATT.IND."
1880 PRINTCHR$(154);"                  1  IMPERFETTO ATT.IND."
1890 PRINTCHR$(156);"                  2  FUTURO 1 ATT.IND."
1900 PRINTCHR$(154);"                  3  PRESENTE PASS.IND."
1910 PRINTCHR$(156);"                  4  IMPERFETTO PASS.IND."
1920 PRINTCHR$(154);"                  5  FUTURO 1 PASS.IND."
1930 PRINTCHR$(156);"                  6  PRESENTE ATT.CONG."
1940 PRINTCHR$(154);"                  7  IMPERFETTO ATT.CONG."
1950 PRINTCHR$(156);"                  8  PRESENTE PASS.CONG."
1960 PRINTCHR$(154);"                  9  IMPERFETTO PASS.CONG."
1970 PRINT:PRINTCHR$(156);"PERSONA     :  1-3 SINGOLARE"
1980 PRINTCHR$(154);"                  4-6 PLURALE":PRINT
1990 PRINTCHR$(150);"CONIUG.:";
2000 GETKJ$:IFKJ$=""THEN2000
2010 KJ=ASC(KJ$):IFKJ<49ORKJ>53THEN2000
2020 KJ=VAL(KJ$)
2030 PRINTKJ;"  FORMA:";
2040 GETFM$:IFFM$=""THEN2040
2050 FM=ASC(FM$):IFFM<48ORFM>57THEN2040
2060 FM=VAL(FM$)
2070 PRINTFM;"  PERSONA:";
2080 GETPR$:IFPR$=""THEN2080
2090 PR=ASC(PR$):IFPR<49ORPR>54THEN2080
2100 PR=VAL(PR$)
2110 PRINTPR;CHR$(152)
2120 PRINTTAB(8);X;TAB(18);Z-1;TAB(30);P
2130 PRINTCHR$(145);CHR$(129);
2140 IFKJ=XTHENPRINT"GIUSTO";TAB(11);:GOTO2160
2150 PRINTCHR$(5);"FALSO";CHR$(129);TAB(11);
2160 IFFM=Z-1THENPRINT"GIUSTO";TAB(23);:GOTO2180
2170 PRINTCHR$(5);"FALSO";CHR$(129);TAB(23);
2180 IFPR=PTHENPRINT"GIUSTO";:GOTO2200
2190 PRINTCHR$(5);"FALSO";CHR$(129);
2200 GETR$:IFR$=""THEN2200
2210 IFR$="S"THEN1760
2220 IFR$<>"N"THEN2200
2230 RUN
```

Istogrammi

Lunghezza del programma: 360 byte e circa 1 Kb. Linguaggio di programmazione: Assembler e BASIC. Accessori: registratore.

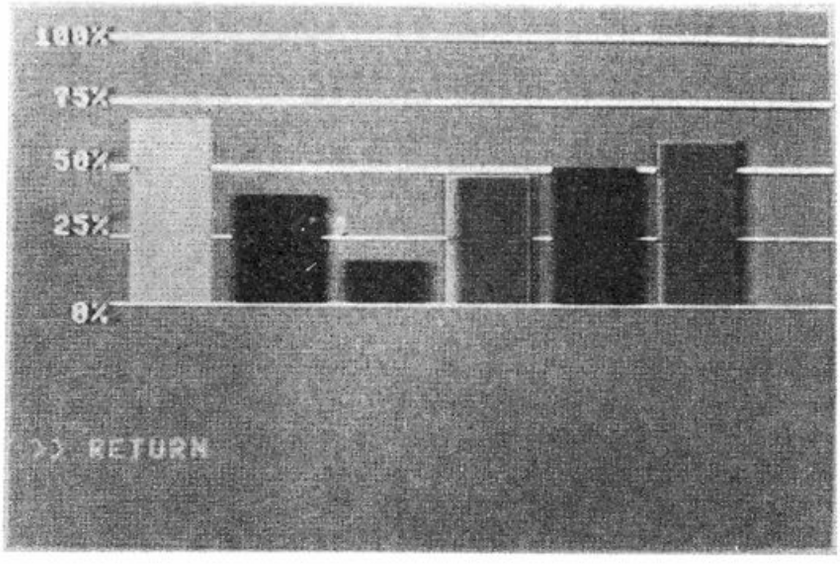
Questo programma in codice macchina permette di rappresentare su video le percentuali in forma di istogramma, con possibilità di scelta del numero, ampiezza, colore delle colonne e colore dello sfondo. I parametri per le variabili vengono trasmessi tramite istruzioni POKE da un programma BASIC, che costituisce un sottoprogramma di "Istogrammi". Sempre dal programma BASIC vengono determinate le percentuali da rappresentare e registrare in aree riservate di memoria.

TRASMISSIONE DEI VALORI

I seguenti indirizzi di memoria della zero-page trasmettono i valori per il programma in codice macchina:
Z = numero delle colonne POKE 169,Z (per un masimo di 17 colonne)
B = ampiezza delle colonne POKE 251,B (ampiezza da 1 a 34)
F = colore e scala POKE 182,F
Le percentuali e i colori delle varie colonne vengono registrati in un secondo buffer su cassetta:
P1 = 1. percentuale
P2 = 2. percentuale e così via
POKE 828,P1
POKE 828,P2 e così via.
F1 = colore della prima colonna
F2 = colore della seconda colonna e così via
POKE 928,F1
POKE 928,F2 e così via.
Una volta che il programma BASIC ha trasmesso tutti i valori necessari, con SYS 52913 si chiama il programma in codice macchina. Prima però bisogna cancellare lo schermo, per evitare che il grafico si mescoli alla videata precedente. In pochi secondi l'istogramma viene visualizzato. L'indirizzo iniziale è 52913, perché il programma in codice macchina è stato memorizzato alla fine dell'area RAM libera, da $CE98 a $CFFF. Altri programmi in codice macchina, il cui indirizzo inziale è solitamente $C000 (decimale: 12 * 4096), non vengono danneggiati dall'istogramma se l'area da CE98 è libera. Per i commenti all'istogramma sono disponibili 10 linee, in cui si può inserire un testo da un programma BASIC.

RAPPRESENTAZIONE DELL'ISTOGRAMMA

Gli istogrammi vengono generati con i simboli della grafica a blocchi. La colonna che rappresenta il



Rappresentazione dell'istogramma sullo schermo

100% è costituita da 100 linee video, mentre il 76% è rappresentato da 76 linee. Questo si ottiene componendo un numero opportuno di simboli grafici, considerando che un carattere ha un'altezza di otto linee. La grafica a blocchi comprende tutti i caratteri di altezza da una ad otto linee. Sovrapponendo un numero esatto di caratteri, si può rappresentare qualsiasi valore tra 1 e 100.
Percentuale: 8 = numero di un blocco completo
Il resto della divisione indica quante linee mancano alla cima della colonna:
per esempio 85% 85 : 8 = 10 con resto 5
Quindi la colonna dell'85% sarà formata da 10 blocchi completi e da un blocco di 5 linee. Il programma in codice macchina trova il blocco finale nella tabella alla fine del programma (da CFF8).

TRATTAMENTO

1. Introdurre e memorizzare il programma di caricamento BASIC.
2. Lanciarlo con RUN. Se il computer si annuncia con "Il programma in codice macchina è caricato", significa che l'input era corretto e il programma gira. Il programma in codice macchina si trova nell'area di memoria che inizia a $CE98.
3. Il programma di caricamento a questo punto può venire cancellato. Quest'operazione non ha alcun effetto sul programma in codice macchina caricato, perché l'area da $C000 non può essere cancellata da NEW né occupata da un programma BASIC.
4. Adesso caricate il programma BASIC, che viene utilizzato come sottoprogramma di "Istogrammi". Se i parametri vengono trasmessi come indicato, il programma in codice macchina può essere chiamato con SYS 52913.
5. Una soluzione particolarmente elegante è quella di appendere il programma di caricamento BASIC ad un programma già esistente. Prima dell'avvio del programma principale, tramite un salto ad un sottoprogramma per il caricamento del programma in codice macchina, quest'ultimo viene caricato. Dopo il salto di ritorno il programma principale prende il via e in questo modo si evita un doppio caricamento del sottoprogramma e del programma principale.
Il programma "demo" serve a darvi un'idea dell'aspetto di un programma BASIC per il controllo di "Istogrammi". Alla linea 130 il numero massimo delle colonne viene calcolato in funzione della loro ampiezza:
numero delle colonne = INT (35/(ampiezza + 1))

Thilo v. Stillfied

```
140 REM ****************************************
150 REM *         C 64 - ISTOGRAMMI              *
160 REM * CARICATORE BASIC DEL PRORAMMA          *
170 REM * IN CODICE MACCHINA PER LA GENERAZIONE  *
180 REM * DI ISTOGRAMMI PERCENTUALI.             *
190 REM * SCALA:      0 - 25 - 50 - 75 - 100 %   *
200 REM *                                        *
210 REM * LUNGHEZZA: 359 BYTES   $CE98 - $CFFF   *
220 REM *                                        *
230 REM * POKE 169,Z   NUMERO DELLE COLONNE      *
240 REM * POKE 251,B   AMPIEZZA DELLE COLONNE    *
250 REM * POKE 182,F   COLORE DELLA SCALA        *
260 REM * POKE 828,P1     1.PERCENTUALE          *
270 REM * POKE 928,F1     COLORE COLONNA 1       *
280 REM * POKE 829,P2     2.PERCENTUALE          *
290 REM * POKE 929,F2     COLORE COLONNA 2       *
300 REM * ... ECC. ...                           *
310 REM * PRINT CHR$(147) PULISCE IL VIDEO       *
320 REM * SYS 52913       CHIAMATA               *
330 REM * (C) THILO V. STILLFRIED                *
340 REM * (C) TECNICHE NUOVE                     *
350 REM *     VIA MOSCOVA 46/9A MILANO           *
360 REM ****************************************
370 :
380 :
```

```
390 :
400 FOR I = 52888 TO 53247
410 ::      READ A
420 ::      IF A=0 THEN N=N+1
430 ::      P=P+A
440 ::      POKE I,A
450 NEXT I
460 :
470 :
480 IF N <> 17    THEN PRINT" ERRORE NELLE NULL-DATA !":END
490 IF P <> 50850 THEN PRINT" ERRORE NEI DATA!":END
500 PRINT"IL PROGRAMMA IN LM E' CARICATO!"
510 PRINT"CANCELLARE IL PROGRAMMA BASIC CON 'NEW'!!"
520 :
530 :
540 DATA 049,048,048,037,064,032,055,053,037,070,032,053,048,037,082,032
550 DATA 050,053,037,100,032,032,048,037,099,169,000,133,253,169,041,133
560 DATA 171,169,004,133,172,169,152,133,254,169,206,133,255,162,000,160
570 DATA 000,177,254,145,171,200,232,224,004,208,246,162,000,177,254,145
580 DATA 171,200,232,224,035,208,248,024,165,171,105,120,144,002,230,172
590 DATA 133,171,165,253,201,003,208,011,024,165,171,105,040,144,002,230
600 DATA 172,133,171,024,165,254,105,005,144,002,230,255,133,254,160,000
610 DATA 230,253,165,253,201,005,208,181,169,000,133,178,169,216,133,179
620 DATA 160,000,165,182,145,178,200,192,000,208,249,230,179,165,179,201
630 DATA 220,208,239,169,000,133,254,169,014,133,171,133,167,169,006,133
640 DATA 172,133,168,162,000,160,000,185,160,003,133,174,185,060,003,133
650 DATA 255,074,074,074,133,252,010,010,010,133,173,165,255,056,229,173
660 DATA 133,253,152,072,165,252,201,000,240,010,169,224,133,170,032,146
670 DATA 207,056,176,004,200,032,220,207,165,171,024,101,251,144,002,230
680 DATA 172,133,171,024,105,001,144,002,230,172,133,171,133,167,165,172
690 DATA 133,168,104,168,200,196,169,208,174,096,169,000,133,254,162,000
700 DATA 160,000,032,232,207,165,170,145,167,165,167,024,105,001,144,002
710 DATA 230,168,133,167,232,228,251,208,231,165,254,197,252,176,048,162
720 DATA 000,165,167,056,233,001,176,002,198,168,133,167,232,228,251,208
730 DATA 240,165,167,056,233,040,176,002,198,168,133,167,230,254,165,254
740 DATA 197,252,208,186,164,253,185,248,207,133,170,192,000,208,175,096
750 DATA 165,167,133,176,165,168,024,105,212,133,177,165,174,145,176,096
760 DATA 032,100,111,121,098,248,247,227
```

```
100 REM * ISTOGRAMMI DEMO *
110 :
120 :
130 POKE 169,6     :REM 6 COLONNE
140 POKE 251,4     :REM AMPIEZZA DI 4 CAMPI
150 POKE 182,1     :REM SCALA IN BIANCO
160 FORI=0 TO 5    :REM DETERMINAZIONE DI...
170 P=INT(100*RND(1)+1)
180 F=INT(15*RND(1)+1)
190 POKE 828+I,P   :REM PERCENTULI E..
200 POKE 928+I,F   :REM COLORE DELLE COLONNE
210 NEXT
220 PRINT"␣"
230 SYS52913
240 PRINT"                          >> RETURN"
250 GETA$:IFA$=""THEN250
260 GOTO130
```

Scritte scorrevoli

Lunghezza del programma: 24000 byte. Linguaggio di programmazione: BASIC. Accessori: registratore o drive, monitor o televisore a colori per proiettare la scritta.

Questo programma permette di genere delle scritte scorrevoli di contenuto a piacere ed è molto adatto a scopi informativi e pubblicitari.

SETTORI APPLICATIVI

Stand di fiere ed esposizioni, negozi di apparecchi radio-televisivi, supermercati, banche, computer shop, agenzie di viaggi, grandi magazzini, artigiani, garage, saloni di bellezza, negozi di giocattoli. L'installazione dell'hardware e del programma è indicata anche per i negozi più piccoli: il C64 potrebbe servire per la pubblicità durante il giorno e come sistema d'allarme di notte.
L'effetto di richiamo è esercitato dallo scorrimento su video di testi e disegni in caratteri colorati, da destra a sinistra, che attirano l'attenzione dei passanti. Dal momento che il programma può generare anche grafici piuttosto grossi, la scritta risulta leggibile anche a grande distanza.

STRUTTURA DEL PROGRAMMA

Il segmento di programma per la generazione della scritta comprende le seguenti funzioni:

1. input dei dati
2. caricamento da disco
3. caricamento da cassetta
4. memorizzazione su disco
5. memorizzazione su cassetta
6. avvio della scritta
7. protezione IN/OFF
8. modifica dei dati
9. cancellazione dei dati
10. fine del programma

Dal momento che i nomi delle funzioni sono abbastanza chiari, spiegheremo soltanto la funzione "protezione". Quando la protezione è attivata, la scritta scorrevole si ferma soltanto allo spegnimento del computer; se la protezione è invece disattivata, la velocità della scritta può essere modificata o si può tornare al menù. La funzione di protezione è consigliabile per evitare che i passanti modifichino o interrompano la scritta, cosa che può succedere soprattutto durante le fiere, negli stand o nei reparti di supermercati.

POSSIBILITÀ

Per quanto riguarda la scritta, esiste la possibilità di far scorrere contemporaneamente su video ben sei diverse linee complete di testo. L'ordine delle linee è programmabile e per ogni linea si può scegliere la dimensione (4, 8 o 16) e il tipo dei caratteri (tra 10 tipi); è possibile anche rappresentare alcuni caratteri di altezza doppia. Inoltre si può scegliere il colore di ogni carattere, dello sfondo di ogni linea, del video e del margine e i caratteri possono essere in reverse. Il testo della scritta può avere una lunghezza di 256 caratteri per linea, per un totale di 1.536 caratteri. Utilizzando i simboli grafici si possono generare anche disegni, passando dal modo maiuscole al modo grafico. Alla fine di ogni scritta il programma ritorna automaticamente al primo carattere.

PARTICOLARITÀ

Il programma per generare scritte scorrevoli presenta le seguenti caratteristiche:
— leggibilità grazie alla particolare organizzazione del video, scelta mirata dei colori, rappresentazione grafica delle funzioni, controllo via menù.
— facilità d'uso con introduzione dell'utente, segnalazione degli errori, sicurezza e input facilitato. La

velocità della scritta per esempio viene controllata mediante i tasti L (lento) e S (svelto).
— grande varietà di dati programmabili per permettere la generazione di scritte personalizzate.

Nils Rode

```
4 :
5 :
6 :
7 :
10 REM ------------------------------------------------------------------------
20 REM -              * * * L A U F S C H R I F T * * *                        -
30 REM -                                                                       -
40 REM -                (C)  COPYRIGHT   NILS RODE  1984                       -
50 REM ------------------------------------------------------------------------
51 :
52 :
53 :
54 :
60 GOTO1000
98 :
99 :
100 REM ** CORNICE ** ---------------------------------------------------------
101 :
102 :
110 PRINT"■┌──────────────────────────────────────┐"
120 FORA=0TO21:PRINT"│"SPC(36)"│":NEXT
130 PRINT"└──────────────────────────────────────┘■":RETURN
138 :
139 :
140 REM ** EMISSIONE ** ------------------------------------------------------
141 :
142 :
150 POKE53265,PEEK(53265)AND239:RETURN
158 :
159 :
160 REM ** IMMISSIONE ** -----------------------------------------------------
161 :
162 :
170 POKE53265,PEEK(53265)OR16:RETURN
178 :
179 :
180 REM ** CHIP SPECIAL ** ----------------------------------------------------
181 :
182 :
190 GOSUB100
200 PRINT"■■■     *** SCRITTE SCORREVOLI ***     "
210 PRINT"■"Q1$:RETURN
218 :
219 :
220 REM *** NON MODIFICARE LE LINEE REM *** ----------------------------------
221 :
222 :
230 SYS52713
240 FORA=685TO707:POKEA,0:NEXT:POKE708,1
250 D6=9:D3=6:FA=1:FORA=1TO6:POKE696+A,128:POKE690+A,30:POKE684+A,0:G(A)=4
255 F2(A)=0:NEXT
260 POKE706,16:POKE705,16:RETURN
298 :
299 :
300 REM ** CORNICE ** ---------------------------------------------------------
```

```
301 :
302 :
310 PRINT"▮▮▮▮▮▮┌──────────────────────────────────────┐"
320 PRINT"▮▮▮▮▮▮|▮▮  *** SCRITTE SCORREVOLI ***  ▮▮|"
330 PRINT"▮▮▮▮▮▮├──────────────────────────────────────┤"
340 PRINT"▮▮▮"CHR$(142):FORB=0TO18:PRINTQ8$:NEXT
350 PRINT"▮▮▮▮▮├───────────── ▮ F7 ▮ = RITORNO ──┤"
400 PRINTTAB(5)" └──────────────────────────────────┘▮":RETURN
408 :
409 :
410 REM ** GENERAZIONE LINEE DI TESTO ** ------------------------------------------
411 :
412 :
420 FORT=ATOB:IFT/5<>INT(T/5)ANDT>0THENA$="   ":GOTO440
430 A$=STR$(T):IFLEN(A$)=3THENA$=RIGHT$(A$,LEN(A$)-1)
440 IFC=0THENPRINT"▮"A$;:GOTO460
450 POKE646,C:PRINT"▮"A$;
460 PRINT"-";:IFD=0THENPRINT"   ";:GOTO480
465 IFT=ATHENPRINT" ┐";:GOTO480
470 IFT=BTHENPRINT" ┘";:GOTO480
475 PRINT" |";
480 PRINTTAB(37)" ":NEXT:RETURN
518 :
519 :
520 REM ** TRASMISSIONE COLORI ** ----------------------------------------------
521 :
522 :
530 F$="":F=0
540 POKE204,0:GETA$:IFA$=""THEN540
550 POKE204,1:IFASC(A$)=13ORA$="▮"ORA$="↑"THENRETURN
555 IFASC(A$)=20ANDF>0THEN610
560 IF(A$<"A")OR(A$>"Z")OR(F>9)THEN540
565 IFM=1THENPRINTF$(16)"▮▮▮▮▮▮▮▮▮▮▮▮";:M=0
570 F$=F$+A$:F=F+1:PRINTA$;
580 FORA=0TO15
590 IFF$<>F$(A)THENNEXT:GOTO540
600 PRINT".":RETURN
610 F$=LEFT$(F$,F-1):F=F-1:PRINT".▮▮.▮▮";:GOTO540
618 :
619 :
620 REM ** ASSEGNAZIONE COLORI ** ----------------------------------------------
621 :
622 :
630 PRINT"▮";:B=-1:F=PEEK(705)AND15:P=PEEK(697):IFP=128THEN690
640 IFP=0THEN660
650 A=0:B=P-1:C=F:D=0:GOSUB410
660 FORV=1TOZ:A=P:B=A+G(V)-1:C=PEEK(684+V)AND15:D=1:GOSUB410:IFV=ZTHEN685
665 A=B+1:P=PEEK(697+V):IFP=128THEN685
670 B=P-1:IFA=PTHEN680
675 C=F:D=0:GOSUB410
680 NEXTV
685 IFB=23THEN695
690 A=B+1:B=23:C=F:D=0:GOSUB410
695 POKE53280,PEEK(706)AND15:RETURN
698 :
699 :
700 REM ** ATTIVAZIONE PROTEZIONE ** -------------------------------------------
701 :
702 :
```

```
710 GOSUB140:GOSUB180:PRINT""SPC(22)""
711 FORA=0TO19:PRINTSPC(22)"|":NEXT:PRINTSPC(22)"⊥"
720 PRINTQ4$
730 PRINTSPC(24)"                    "
740 PRINTSPC(24)"                    "
750 PRINTSPC(24)"  SICUREZZA "
760 PRINTSPC(24)"     SI      "
770 FORA=0TO10:PRINTSPC(24)"                 ":NEXT
780 PRINTSPC(24)"     NO      "
790 PRINTSPC(24)"                    "
800 PRINTSPC(24)"                    "
810 A=15:IFPEEK(707)>0THENA=9
820 PRINTLEFT$(Q6$,A)SPC(26)Q9$
830 A$="SI":IFA=9THENA$="SI"
840 RETURN
898 :
899 :
900 REM ** SCELTA DEI COLORI ** -----------------------------------------------
901 :
902 :
910 PRINTLEFT$(Q6$,16)
920 PRINTTAB(6)"NERO BIANCO ROSSO TURCHESE VI-"
930 PRINTTAB(6)"OLA VERDE BLU GIALLO ARANCIO  "
940 PRINTTAB(6)"BRUNO CROSSO SGRIGIO GRIGIO"
950 PRINTTAB(6)"CVERDE CBLU CGRIGIO     ":RETURN
997 :
998 :
999 :
1000 REM ** TITOLAZIONE ** --------------------------------------------------
1001 :
1002 :
1010 POKE53281,0:POKE53280,0:POKE650,128
1020 POKE53270,PEEK(53270)AND247:POKE53265,PEEK(53265)AND247
1030 POKE53270,(PEEK(53270)AND248)+7:POKE53265,(PEEK(53265)AND248)+7
1040 GOSUB140:GOSUB100
1050 PRINTCHR$(142)CHR$(8)SPC(49)"                          "
1060 PRINTSPC(9)"                     "
1070 PRINTSPC(9)"                        "
1080 PRINTSPC(9)"                        "
1090 PRINTSPC(9)"                     "
1100 PRINTSPC(9)"                     "
1110 PRINTSPC(43)"                         |      "
1120 PRINT"                                         "
1130 PRINT"                                        "
1140 PRINT"                                           "
1150 PRINT"                                          "
1160 PRINT"                                          "
1170 PRINTSPC(57)"DI"
1171 PRINTSPC(12)"                 "
1172 PRINT"--------                    --------"
1173 PRINTSPC(12)"     |        | |    "
1174 PRINTSPC(12)"          "|"|"":GOSUB160
1175 :
1190 DIMF$(16),G(6),GR(6),Q$(3),F2(6),F3(6)
1200 FORA=0TO16:READF$(A):NEXT
1210 Q1$="├──────────────────────────────────────────┤"
1220 Q2$="├───────────────────────────────────┤"
1230 Q3$=""+Q2$+"SCELTA :─────────"
```

```
1240 Q4$="":Q5$="LINEA TESTO "
1250 QS$=CHR$(13)+"────────────────────────────────────────────────────────────"
1260 Q$(0)=".......":Q$(1)="COMPRESSO":Q$(2)="NORMALE":Q$(3)="ESPANSO"
1270 Q6$=""
1280 FORA=0TO4:READC:Q8$=Q8$+"|":FORB=0TO29:Q8$=Q8$+CHR$(C):NEXT
1290 Q8$=Q8$+"|"+CHR$(13)+"□":NEXT
1310 Q9$="          ┌─────┐"
1320 Q9$=Q9$+"├─────┤└─────┘         "
1330 POKE43,244
1340 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
1350 FORA=0TO25:READX
1360 FORC=0TO7:POKE2048+A*8+C,PEEK(53248+C+8*X):NEXTC:NEXTA
1370 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
1380 FORA=52224TO52840:READB:POKEA,B:NEXT:GOSUB220
1497 :
1498 :
1499 :
1500 REM ** SCELTA ** -------------------------------------------------------------
1501 :
1502 :
1510 POKE53265,PEEK(53265)AND191
1515 POKE53272,21:POKE53280,0:POKE53281,0:GOSUB140:GOSUB180
1520 PRINT"SCEGLIETE PREGO!:        ":PRINTQ1$""
1530 FORA=0TO4
1540 PRINT"┌┬──────────────┐  ┌┬──────────────┐"
1550 PRINT"| |"SPC(12)"|  | |"SPC(12)"|"
1560 PRINT"└┴──────────────┘  └┴──────────────┘"
1570 NEXT
1580 PRINT""
1590 PRINT"1INPUT DATI    6AVVIO   "
1600 PRINT"2LOAD:    DISK7SICUREZZA"
1610 PRINT"3LOAD:    CASS8MODIFICA"
1620 PRINT"4SAVE:    DISK9CANCELLA"
1630 PRINT"5SAVE:    CASS0FINE":GOSUB160
1640 GETA$:IFA$=""THEN1640
1650 IFA$="0"THEN9950
1660 ON VAL(A$) GOTO 2000,4400,4000,4800,4200,6000,5000,2000,9900
1670 GOTO1640
1997 :
1998 :
1999 :
2000 REM ** INPUT DATI ** ---------------------------------------------------------
2001 :
2002 :
2010 GOSUB140:GOSUB300:GOSUB620:GOSUB160
2020 IFZ>0THEN2082
2030 GOSUB160:PRINTQ4$" QUANTE LINEE DI TESTO ? "Z"";
2031 POKE204,0:GETA$:IFA$=""THEN2031
2032 POKE204,1:IFA$=""THEN1500
2035 IFASC(A$)=13ANDZ>0THEN2080
2040 A=VAL(A$):IFA=0THEN2031
2050 IFA<7THEN2070
2060 PRINT:PRINTQ2$SPC(49)"MASSIMO 6 LINEE DI TESTO     "
2065 GOTO2030
2070 IFA<ZTHEN2072
2071 G=0:FORB=ATO1STEP-1:IFPEEK(696+B)=128THENG=G+G(B):NEXT
2072 IFPEEK(698)=128THENIFG<25THEN2075
2073 IFPEEK(696+B)+G>24THEN2030
2075 IFA>=ZTHEN2077
```

```
2076 FORB=A+1TOZ:POKE696+B,128:G(B)=4:POKE684+B,0:POKE690+A,30:F2(B)=0:NEXT
2077 Z=A:IFPEEK(704)=0THENGOSUB220
2078 POKE704,Z:PRINT"■"Z:GOSUB620:S=0:POKE204,1
2080 IFS=2ORS=6THEN2085
2081 GOSUB340
2082 PRINTQ3$
2085 N=1:ON S GOTO2100,2300,2500,2800,2950,3100
2099 :
2100 REM DIMENSIONI
2101 :
2110 PRINTSPC(6)"LE 3 DIMENSIONI DEI CARATTERI:"
2120 PRINTSPC(7)"■1. COMPRESSO : 4 LINEE "
2130 PRINTSPC(7)"2. NORMALE    : 8 LINEE "
2140 PRINTSPC(7)"3. ESPANSO    :16 LINEE "
2150 PRINTQ4$TAB(6)"□□■DIMENSIONI CARATTERE"T  QS$Q4$
2160 FORA=1TOZ:PRINT"■"Q5$A" : "Q$(GR(A)):NEXT
2180 PRINTLEFT$(Q6$,6+N)TAB(25);
2190 POKE204,0:GETA$:IFA$=""THEN2190
2195 POKE204,1:IFA$="■"THEN1500
2200 IFA$="N"THENB3=2:B1=8:B2=0:GOTO2265
2210 IFA$="C"THENB3=1:B1=4:B2=128:GOTO2265
2220 IFA$="E"THENB3=3:B1=16:B2=1:GOTO2265
2230 IFASC(A$)=13ANDGR(N)>0THENPRINTQ$(GR(N)):GOTO2280
2240 IFA$<>"↑"THEN2190
2245 IFN=1THENGOSUB340:GOTO2030
2250 PRINTQ$(GR(N)):N=N-1:GOTO2180
2260 IFS>1THENS=S-1:GOTO2080
2265 C=0:FORA=1TOZ:C=C+G(A):NEXT:C=C+B1-G(N):IFC>24THEN2190
2266 P=PEEK(696+N):IFP=128THEN2270
2267 IFN<ZTHEN2269
2268 IFP+B1>24THEN2190
2269 IFP+B1>PEEK(697+N)THEN2190
2270 G(N)=B1:PRINTQ$(B3):IFGR(N)>0ANDGR(N)<>B3THENGOSUB620
2275 GR(N)=B3:POKE690+N,(PEEK(690+N)AND30)ORB2
2280 N=N+1:IFN>ZTHENS=2:GOTO2080
2290 GOTO2180
2299 :
2300 REM TIPO DI CARATTERE
2301 :
2303 FORA=1TOZ:IF(PEEK(690+A)AND128)<128THEN2309
2305 NEXTA:IFA$="↑"THENS=1:GOTO2080
2306 S=3:GOTO2080
2309 GOSUB340:PRINTQ3$
2310 PRINTTAB(6)"SONO DISPONIBILI 10 TIPI DI  "SPC(11)"CARATTERI:"
2330 PRINTTAB(6)"■ 1 ■ ■● ■ 2 ■ ■○■ 3 ■ ■* ■ 4 ■ ■+ ■ 5 ■ ■♦"
2340 PRINTTAB(6)"■ 6 ■ ■※ ■ 7 ■■■ 8 ■ ■X ■ 9 ■ ■# ■ 0 ■■ ■"
2345 PRINTQ4$"□□"TAB(6)"■TIPO DI CARATTERE"QS$Q4$
2350 FORN=1TOZ:PRINTQ5$N" : ";:P=(PEEK(690+N)AND30):GOSUB2491:NEXTN:N=1
2380 PRINTQ4$
2390 PRINTTAB(24);:IFGR(N)=1THENA=13:P=0:GOTO2430
2400 POKE204,0:GETA$:IFA$=""THEN2400
2410 POKE204,1:IFA$="■"THEN1500
2420 A=ASC(A$):IFA>47ANDA<58THEN2470
2425 P=PEEK(690+N)AND30
2430 IFA=13ANDP<30THENGOSUB2491:GOTO2480
2440 IFA$<>"↑"THEN2400
2450 IFN=1THENS=1:GOTO2080
2460 N=N-1:GOSUB2491:PRINT"□□":IFGR(N)=1THEN2450
2465 GOTO2390
```

```
2470 A=VAL(A$):POKE690+N,(PEEK(690+N)AND225)OR(A*2):PRINT"■"A
2480 N=N+1:IFN>2THENS=3:GOTO2080
2490 GOTO2390
2491 PRINTTAB(24);:IFGR(N)=1THENPRINT"-":RETURN
2492 IFP=30THENPRINT"■":RETURN
2493 PRINT"■"P/2:RETURN
2499 :
2500 REM ORDINAMENTO DEL VIDEO
2501 :
2510 PRINTTAB(6)"INTRODUCETE I NUMERI DI LINEA "SPC(8);
2520 PRINTTAB(6)"E PREMETE IL TASTO"SPC(2)"■■ RETURN ■ ■!"
2525 PRINTQ4$"□□"TAB(6)"■ORDINAMENTO DEL VIDEO"QS$Q4$
2530 FORN=1TO2:PRINT"■■■■■■LA SCRITTA"N;:GOSUB2790:NEXT:N=1
2550 PRINT"■"LEFT$(Q6$,21)"■■"Q5$N"HA"G(N)"■■ LINEE  "
2555 Z2=24:IFN=1THENZ1=0:GOTO2570
2560 Z1=PEEK(696+N-1)+G(N-1)
2570 IFN=2THENZ2=Z2-G(N):GOTO2600
2580 IFPEEK(697+N)<25THENZ2=PEEK(697+N)-G(N):GOTO2600
2590 FORT=NTO2:Z2=Z2-G(T):NEXT
2600 PRINTLEFT$(Q6$,6+N)TAB(18)Z1:PRINT"□"TAB(21);Z2
2605 PRINT"□"TAB(18)"(■■-■■) : ";
2610 IFZ1<>Z2THEN2620
2611 IFA$<>"↑"THENA=Z1:GOTO2750
2612 POKE204,0:GETA$:IFA$=""THEN2612
2613 POKE204,1:IFA$="↑"ANDN=1THENS=2:GOTO2080
2615 A=Z1:GOTO2750
2620 R=0:R$=""
2630 POKE204,0:GETA$:IFA$=""THEN2630
2640 POKE204,1:A=ASC(A$):IFA>47ANDA<58ANDR<2THEN2720
2650 IFA$="■"THEN1500
2660 IFA=13THEN2730
2670 IFA=20ANDR>0THEN2710
2680 IFA$<>"↑"ORR>0THEN2630
2690 IFN=1THENS=2:GOTO2080
2700 GOSUB2790:GOSUB2792:PRINT"□□":N=N-1:GOTO2550
2705 GOTO2550
2710 PRINT" ■■ ■■";:R=R-1:R$=LEFT$(R$,LEN(R$)-1):GOTO2630
2720 R$=R$+A$:R=R+1:PRINTA$;:GOTO2630
2730 IFR=0THEN2770
2740 PRINT" ";:A=VAL(R$):IFA<Z1ORA>Z2THEN2550
2750 POKE696+N,A:GOSUB2790:GOSUB2792:GOSUB620
2760 N=N+1:IFN>2THENS=4:GOTO2080
2765 GOTO2550
2770 IFPEEK(696+N)<24THENGOSUB2790:GOSUB2792:GOTO2760
2775 GOTO2550
2790 PRINTCHR$(13)"□"TAB(27);:IFPEEK(696+N)>24THENPRINT"     ":RETURN
2791 PRINTPEEK(696+N)"  ":RETURN
2792 PRINT"□"TAB(18)"       ":RETURN
2799 :
2800 REM INPUT COLORE 1
2801 :
2805 GOSUB900
2810 PRINTQ4$"■■■■■■■■COLORE VIDEO:    "F$(PEEK(705))
2811 PRINT"■■■■■■■■COLORE BORDO:    "F$(PEEK(706))Q4$SPC(63);
2815 M=0:IFPEEK(705)<16THENM=1
2820 GOSUB520:IFA$="■"THEN1500
2830 IFA$="↑"THENS=3:GOTO2080
2840 IFASC(A$)<>13THEN2850
```

```
2841 IFPEEK(705)>15THEN2810
2845 PRINTCHR$(13)"□"TAB(23)F$(PEEK(705))LEFT$(F$(16),11-LEN(F$(PEEK(705))))
2846 GOTO2860
2850 POKE705,A:GOSUB620
2860 PRINTLEFT$(Q6$,9)SPC(23)"◣";
2865 M=0:IFPEEK(706)<16THENM=1
2870 GOSUB520:IFA$="▌"THEN1500
2880 IFA$<>"↑"THEN2890
2881 PRINTCHR$(13)"□"TAB(23);
2882 PRINTF$(PEEK(706))LEFT$(F$(16),11-LEN(F$(PEEK(706))));:GOTO2810
2890 IFASC(A$)=13ANDPEEK(706)<16THENS=5:GOTO2080
2900 IFASC(A$)=13THEN2865
2910 POKE706,A:POKE53280,A:S=5:GOTO2080
2949 :
2950 REM INPUT COLORE 2
2951 :
2955 PRINTQ4$"□□"TAB(6)"◣COLORE SFONDO:    "Q5$Q4$
2960 FORA=1TOZ:PRINT"▌"Q5$A" : ";F$(PEEK(684+A)AND15):NEXT
2970 GOSUB900
2980 PRINT"◣"LEFT$(Q6$,6+N)TAB(24);:P=PEEK(684+N)AND15
2990 M=1
3000 GOSUB520:IFA$="▌"THEN1500
3010 IFA$<>"↑"THEN3040
3020 IFN=1THENS=4:GOTO2080
3030 N=N-1:PRINTCHR$(13)"□"TAB(24)F$(P):GOTO2980
3040 IFASC(A$)=13THENPRINTCHR$(13)"□"TAB(24)F$(P):GOTO3070
3050 F2(N)=A:F1=1:POKE685,(PEEK(685)AND15)OR64:POKE53282,F2(1):IFZ=1THEN3060
3051 FORB=2TOZ:FORC=1TOF1:IF(PEEK(53281+C)AND15)=F2(B)THEN3054
3052 NEXTC:IFF1=3THEN2970
3053 F1=F1+1:POKE53281+F1,F2(B):C=F1
3054 POKE684+B,F2(B)OR(C*64):NEXTB
3060 POKE684+N,(PEEK(684+N)AND192)ORF2(N):GOSUB620
3070 N=N+1:IFN>ZTHENS=6:GOTO2080
3080 GOTO2980
3099 :
3100 REM INPUT DEL TESTO
3101 :
3110 GOSUB340:POKE78,0:POKE182,0
3150 PRINTQ4$TAB(6)"□▌COLORE SCRITTE:";:POKE646,FA:IF FA>0THEN3170
3155 PRINT"▀    NERO    ▀"
3160 PRINTSPC(21)"□□▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀":PRINTSPC(21)"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀":GOTO3199
3170 A$="▀                ":B=6-LEN(F$(FA))/2
3180 PRINT"▀"RIGHT$(A$,B)F$(FA)RIGHT$(A$,12-LEN(F$(FA))-INT(B))
3190 PRINT"□□"SPC(21)A$SPC(68)A$
3199 PRINTLEFT$(Q3$,9)Q2$:FORA=1TOZ:PRINT:NEXT:PRINTQ2$
3200 PRINTSPC(6)"▀▀▀ ↑ ◣ CONVERSIONE AMPIEZZA "
3201 PRINTSPC(6)"▀▀▀ ← ◣ FINE IMMISSIONE CARATTERI"
3202 PRINTSPC(6)"▀▀▀ CLR/HOME ▀▀▀ ESPANSO/COMPRESSO"
3203 PRINTSPC(6)"▀▀▀ INST/DEL ▀▀▀ NEGATIVO/POSITIVO"
3204 PRINTSPC(6)"▀▀▀ F1 ▀▀ ◣  SCORRIMENTO PAGINE"
3206 POKE53272,PEEK(708)*21+(PEEK(708)=0)*(-23):SYS52738
3210 POKE646,FA:POKE211,D3:POKE214,D6:SYS58732
3215 POKE204,0:POKE207,0:GETA$:IFA$=""THEN3215
3220 POKE204,1:IFA$="▌"THEN1500
3230 A=ASC(A$):IFA=64THENFA=(FA+1)AND15:GOTO3150
3240 IF(A>31ANDA<128)OR(A>159ANDA<224)ORA>224THEN3290
3250 IFA=29THEN3310
3260 IFA=17THEN3400
3270 IFA=145THEN3370
```

```
3280 IFA=157THEN3430
3281 IFA=19THEN3500
3282 IFA=20THEN3530
3283 IFA=13ANDD6<Z+8THENPOKE78,0:D3=6:D6=D6+1:GOTO3206
3284 IFA=133THENS=5:GOTO2080
3285 GOTO3215
3290 POKE199,PEEK(182):PRINTA$:A=256*(D6-9)+PEEK(78)+D3-6:B=D6*40+D3
3300 POKE49152+A,PEEK(1024+B):POKE50688+A,PEEK(55296+B)
3310 REM DESTRA
3320 PRINT"■":IFD3<35THEND3=D3+1:GOTO3206
3330 IFPEEK(78)=223THEN3350
3340 POKE78,PEEK(78)+1:GOTO3206
3350 IFD6=Z+8THEN3206
3360 D6=D6+1:D3=6:POKE78,0:GOTO3206
3370 REM SU'
3380 IFD6=9THEN3206
3390 D6=D6-1:GOTO3206
3400 REM GIU'
3410 IFD6=8+ZTHEN3206
3420 D6=D6+1:GOTO3206
3430 REM SINISTRA
3440 IFD3=6THEN3460
3450 D3=D3-1:GOTO3206
3460 IFPEEK(78)=0THEN3480
3470 POKE78,PEEK(78)-1:GOTO3206
3480 IFD6=9THEN3206
3490 D6=D6-1:D3=35:POKE78,223:GOTO3206
3500 REM ESPANSI/COMPRESSI
3510 IFPEEK(708)=1THENPOKE708,0:GOTO3206
3520 POKE708,1:GOTO3206
3530 REM NEGATIVO
3540 IFPEEK(182)=0THENPOKE182,1:GOTO3206
3550 POKE182,0:GOTO3206
3997 :
3998 :
3999 :
4000 REM ** LOAD CASS ** ----------------------------------------------------
4001 :
4002 :
4020 GOSUB9000:P=1:GOSUB9705:GOSUB7200
4030 PRINT"■":GOSUB9050:PRINT"■■PREMETE ■■ PLAY ■■ SUL REGISTRATORE"
4040 REM
4050 F=5
4060 FORA=0TO159STEP40:FORB=55986+ATO55988+A:POKEB,F
4070 P=PEEK(1)AND16:IFP=0THEN4120
4080 NEXT:NEXT:IFF=5THENF=13:GOTO4100
4090 F=5
4100 GETA$:IFA$<>""THEN1500
4110 FORA=0TO350:NEXT:GOTO4060
4120 PRINT"■■":OPEN1,1,0,F$:DD=0:GOTO7000
4197 :
4198 :
4199 :
4200 REM ** SAVE CASS ** ----------------------------------------------------
4201 :
4202 :
4210 GOSUB9000:GOSUB9700:IFF$=""THEN1500
4215 GOSUB7200:PRINT"■":GOSUB9050
```

```
4220 PRINT"■■PREMETE ■■ RECORD ■■ & ■■ PLAY ■ SUL REGISTRATORE"
4230 REM
4240 F=5
4250 FORA=0TO159STEP40:FORB=55983+ATO55988+A:POKEB,F:NEXT:NEXT
4260 IFF=5THENF=13:GOTO4280
4270 F=5
4280 P=PEEK(1)AND16:IFP=0THEN4310
4290 GETA$:IFA$<>""THEN1500
4300 FORA=0TO350:NEXT:GOTO4250
4310 PRINT"■■":OPEN1,1,1,F$:DD=0:GOTO7100
4397 :
4398 :
4399 :
4400 REM ** LOAD DISK ** ----------------------------------------------------
4401 :
4402 :
4410 GOSUB9400
4420 GOSUB9700:IFF$=""THEN1500
4430 GOSUB4791:POKE198,1:POKE631,13:PRINT"■■■■ GOTO4600■";:FF=0
4440 OPEN15,8,15:OPEN1,8,4,F$+",S,R":DD=1:GOSUB7190:IFFF>0THEN4600
4450 GOTO7000
4597 :
4598 :
4599 :
4600 REM ** MESSAGGI ERRORI DISCO ** ----------------------------------------
4601 :
4602 :
4605 CLOSE1:CLOSE15:POKE204,1
4610 IFFF=63THEN4760
4620 IFFF=0THEN4700
4630 IFFF<>62THENPRINT"■■■ERRORE DISCO: "FF$:FORA=0TO3000:NEXT:GOTO1500
4640 REM *** NON TROVATO ***
4650 PRINT"■■■  UN FILE CON QUESTO NOME NON E'     "
4660 PRINT"    PRESENTE SU QUESTO DISCO!"
4670 GOSUB9410:GOTO4420
4700 REM *** PERIFERICA NON PRESENTE ***
4710 PRINT"■■■NON AVETE ACCESO O"
4720 PRINT"COLLEGATO IL DRIVE!"
4740 PRINT"■■■■(PREGO PREMETE UN TASTO !)"
4750 POKE198,0:WAIT198,1:GOTO1500
4760 REM *** FILE ESISTENTE ***
4770 PRINT"■■■  UN FILE CON QUESTO NOME E' GIA'"
4780 PRINT"    PRESENTE SU QUESTO DISCO!"
4790 GOSUB9410:GOTO4820
4791 REM ** CANCELLAZIONE DELLA META' SUPERIORE DEL VIDEO **
4792 PRINT"■":FORA=0TO4:PRINT"                                        ":NEXT:RETUR
N
4797 :
4798 :
4799 :
4800 REM ** SAVE DISK ** ----------------------------------------------------
4801 :
4802 :
4810 GOSUB9400
4820 GOSUB9700:IFF$=""THEN1500
4830 GOSUB4791:POKE198,1:POKE631,13:PRINT"■■■■ GOTO4600■";:FF=0
4840 OPEN15,8,15:OPEN1,8,4,F$+",S,W":DD=1:GOSUB7190:IFFF>0THEN4600
4850 GOTO7100
```

```
4997 :
4998 :
4999 :
5000 REM ** SICUREZZA ** ----------------------------------------------------
5001 :
5002 :
5010 GOSUB700
5140 PRINTQ4$"■QUANDO LA SICUREZZA "
5150 PRINT"■E' ATTIVATA, LA"
5160 PRINT"■SCRITTA SCORREVOLE  "
5170 PRINT"■NON PUO' PIU' ESSERE"
5180 PRINT"■INTERROTTA."
5185 PRINT"■┣━━━━━━━━━━━━━━━━━━━━━━━━┫"
5190 PRINT"■■LA SICUREZZA DEVE   "
5200 PRINT"■ESSERE ATTIVATA?"
5201 PRINT"■┣━━━━━━━━━━━━━━━━━━━━━━━━┫"
5205 PRINT"■■PER FAVORE PREMETE "
5206 PRINT"■■■ S ■ PER  ■'SI'■  O   "
5207 PRINT"■PREMETE ■■ F7 ■ !":GOSUB160
5210 GETA$:IFA$=""THEN5210
5220 IFA$="■"THEN1500
5230 IFA$<>"S"THEN5210
5240 POKE707,0:IFA>9THENPOKE707,128
5250 IFA=9THENFORB=9TO15:GOTO5270
5260 FORB=15TO9STEP-1
5270 PRINTLEFT$(Q6$,B)SPC(26)Q9$:NEXT
5280 FORA=0TO3500:NEXT:GOTO1500
5997 :
5998 :
5999 :
6000 REM ** AVVIO ** -------------------------------------------------------
6001 :
6002 :
6010 GOSUB700:IF684+Z=160RZ=0THEN6300
6020 IFA<15THEN 6200
6030 PRINTQ4$"□□■   CONTROLLATE LA   "
6040 PRINT"■VELOCITA' DELLA     "
6050 PRINT"■SCRITTA SCORREVOLE "
6060 PRINT"■COI TASTI ■■ L ■(LENTO)"
6070 PRINT"■D ■■ S ■(SVELTO)"
6080 REM
6090 PRINT"■■┣━━━━━━━━━━━━━━━━━━━━━━━━┫"
6100 PRINT"■■■CON LO   ■■   SPACE   ■"
6110 PRINT"■■POTETE TORNARE     "
6120 PRINT"■ALLA PARTE       "
6130 PRINT"■DI INPUT."
6140 PRINT"■■┣━━━━━━━━━━━━━━━━━━━━━━━━┫"
6150 PRINT"■■■PREMETE                "
6160 PRINT"■■■■ S ■  PER PARTIRE O   "
6170 PRINT"■■■■ F7 ■ !":GOSUB160
6180 GOTO6400
6200 PRINTQ4$"■■■SE ADESSO PREMETE   "
6210 PRINT"■LA ■■ S PER    "
6220 PRINT"■■AVVIARE, LA SCRITTA "
6230 PRINT"■SCORREVOLE, NON SARA'"
6240 PRINT"■PIU' INTERROTTA."
6250 PRINT"■■┣━━━━━━━━━━━━━━━━━━━━━━━━┫"
6260 PRINT"■■■CON ■■ F7 ■ RITORNATE":GOSUB160
6270 GOTO6400
```

```
6300 PRINTQ4$"■NON AVETE ANCORA     "
6310 PRINT"■FINITO DI INTRODURRE"
6320 PRINT"■I DATI !!!":GOSUB160:FORA=0TO3000:NEXT:GOTO1500
6400 GETA$:IFA$=""THEN6400
6410 IFA$="■"THEN1500
6420 IFA$<>"S"THEN6400
6430 POKE709,PEEK(696+Z)+G(Z)
6431 FORA=685TO690:F3(A-684)=PEEK(A):POKEA,F3(A-684)AND192:NEXT
6440 FORA=710TO715:POKEA,128:NEXT:POKE53265,PEEK(53265)OR64
6445 PRINT"■":POKE646,PEEK(705):FORA=0TO999:PRINT"■ ";:NEXT
6450 POKE53281,PEEK(705):PRINT"■":POKE53280,PEEK(706)
6455 FORA=691TO696:POKEA,PEEK(A)AND191:NEXT
6500 POKE53272,(PEEK(53272)AND240)OR2:SYS52797:PRINT"■"
6505 FORA=685TO690:POKEA,F3(A-684):NEXT
6510 GOTO1500
6998 :
6999 :
7000 REM ** CARICAMENTO ** ----------------------------------------------------
7001 :
7002 :
7005 GOSUB220
7010 FORA=53282TO53284:GET#1,A$:GOSUB7190:IFFF>0THEN4600
7015 POKEA,ASC(A$+CHR$(0)):NEXT
7020 FORA=685TO708:GET#1,A$:GOSUB7190:IFFF>0THEN4600
7025 POKEA,ASC(A$+CHR$(0)):NEXT:Z=PEEK(704)
7030 FORA=0TOZ-1:FORT=0TO255:GET#1,A$:GOSUB7190:IFFF>0THEN4600
7035 P=ASC(A$):GET#1,B$:GOSUB7190:IFFF>0THEN4600
7040 POKE49152+A*256+T,P
7050 POKE50688+A*256+T,ASC(B$+CHR$(0)):IFP<>31THENNEXTT
7060 NEXTA:CLOSE1:CLOSE15
7070 FORA=1TO6:P=PEEK(690+A)AND225:IFP=0THENG(A)=8:GR(A)=2
7075 IFP=1THENG(A)=16:GR(A)=3
7080 IFP=128THENG(A)=4:GR(A)=1
7085 NEXTA
7090 FORA=685TO690:F2(A-684)=PEEK(A)AND15:NEXT
7091 IFDD=0THENGOSUB7200
7092 GOTO1500
7098 :
7099 :
7100 REM ** MEMORIZZAZIONE ** ---------------------------------------------------
7101 :
7102 :
7110 FORA=53282TO53284:PRINT#1,CHR$(PEEK(A));:GOSUB7190:IFFF>0THEN4600
7115 NEXT
7120 FORA=685TO708:PRINT#1,CHR$(PEEK(A));:GOSUB7190:IFFF>0THEN4600
7125 NEXT
7130 FORA=0TOZ-1:FORT=0TO255
7140 P=PEEK(49152+T+A*256):PRINT#1,CHR$(P)CHR$(PEEK(50688+A*256+T));
7145 GOSUB7190:IFFF>0THEN4600
7150 IFP<>31THENNEXTT
7160 NEXTA:CLOSE1:CLOSE15
7170 IFDD=0THENGOSUB7200
7180 GOTO1500
7188 :
7189 :
7190 REM ** LETTURA CANALE ERRORE ** ---------------------------------------
7191 :
7192 :
7193 IFDD=0THENRETURN
```

```
7194 INPUT#15,FF,FF$:RETURN
7198 :
7199 :
7200 REM ** CONTROLLO DI STOP ** -----------------------------------------
7201 :
7202 :
7205 GOSUB9040
7210 P=PEEK(1)AND16:IFP>0THENRETURN
7220 PRINT"...PREMETE ... STOP ... SUL      "
7230 PRINT"...REGISTRATORE!..."
7240 F=5
7250 FORA=0TO159STEP40:FORB=55995+ATO55997+A:POKEB,F
7260 P=PEEK(1)AND16:IFP>0THENRETURN
7270 NEXT:NEXT:FORA=0TO350:NEXT:IFF=5THENF=13:GOTO7250
7280 GOTO7240
8998 :
8999 :
9000 REM ** GRAFICA SU CASSETTA ** --------------------------------------
9001 :
9002 :
9010 GOSUB140:PRINT"..."
9040 PRINT"..."
9050 PRINT"..."
9060 PRINT"..."
9070 PRINT"..."
9080 PRINT"..."
9090 PRINT"..."
9100 PRINT"..."
9110 PRINT"..."
9120 PRINT"..."
9130 PRINT"...  ====>  ..."
9150 PRINT"...  AUTO STOP  ..."
9160 PRINT"..."
9170 PRINT"..."
9180 PRINT"...  •SAVE  ..."
9190 PRINT"...  COMMODORE  ..."
9200 PRINT"..."
9210 PRINT"...  •  > <<  >>  ...  ↑  ..."
9220 PRINT"..."
9230 PRINT"..."
9240 PRINT"..."
9250 PRINT"..."
9260 PRINT"...":GOSUB160:RETURN
9398 :
9399 :
9400 REM ** GRAFICA SU DISCO ** -----------------------------------------
9401 :
9402 :
9405 GOSUB140:PRINT"..."
9410 PRINT"..."
9420 PRINT"...  SINGLE DRIVE  ..."
9430 PRINT"...  FLOPPY  DISK  ..."
9440 PRINT"..."
9450 PRINT"..."
9460 PRINT"..."
9470 PRINT"..."
9480 PRINT"..."
9490 PRINT"... • ... •..."
9500 PRINT"...":GOSUB160:RETURN
```

```
9698 :
9699 :
9700 REM ** TRASMISSIONE NOME DEL FILE ** ------------------------------------------------
9701 :
9702 :
9704 P=0
9705 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■  ▶NOME FILE:                              "
9710 PRINT"                                          ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
9720 F$="":F=0:POKE204,0:POKE198,0
9730 GETA$:IFA$=""THEN9730
9740 IF(ASC(A$)=20)AND(F>0)THEN9790
9745 IFASC(A$)=13THEN9780
9750 IF(A$<" ")OR(A$>"Z")OR(F>15)THEN9730
9760 F$=F$+A$:F=F+1:PRINTA$;:GOTO9730
9780 POKE204,1:PRINT" ":IFF$<>""ORP=1THENRETURN
9782 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■INTRODUCETE UN NOME PER IL FILE !"
9783 PRINT"■■─────────────────────────────────────────────────":FORA=0TO2000:NEXT
9784 P=1:GOTO9705
9790 F=F-1:F$=LEFT$(F$,F):PRINT" ■■■ ■■";:GOTO9730
9897 :
9898 :
9899 :
9900 REM ** CANCELLAZIONE ** ---------------------------------------------------------
9901 :
9902 :
9910 GOSUB140:GOSUB180
9920 PRINT"■■▶ ┌┐":PRINT"■┘CANCELLO TUTTI I DATI ? ■(S/N)":PRINT"■▶ └┘"
9930 GOSUB160
9940 GETA$:IFA$=""THEN9940
9945 IFA$<>"S"THEN1500
9946 RUN
9948 :
9949 :
9950 REM ** FINE ** ------------------------------------------------------------------
9951 :
9952 :
9955 GOSUB140:GOSUB180:PRINT"■■▶ ┌┐ ":PRINT"■■| |VOLETE TERMINARE IL         "
9960 PRINT"■■| | PROGRAMMA? ■(S/N)":GOSUB160
9970 GETA$:IFA$=""THEN9970
9980 IFA$<>"S"THEN1500
9990 SYS64738
9995 :
9996 :
9997 :
9998 :
9999 :
10000 DATA"NERO","BIANCO","ROSSO","TURCHESE","VIOLA"
10005 DATA"VERDE","BLU","GIALLO"
10010 DATA"ARANCIO","BRUNO","CROSSO","SGRIGIO"
10015 DATA"GRIGIO","CVERDE","CBLU"
10020 DATA"CGRIGIO","..........."
18998 :
18999 :
19000 DATA183,196,198,175,32
19001 DATA160,81,87,42,91,90,102,207,86,35,32,124,126,226,108,225,127,251,123
19002 DATA255,97,236,98,254,252,160
19995 :
19996 :
19997 :
```

```
19998 :
19999 :
20000 DATA173,185,2,133,214,32,108,229,32,36,234,166,214,160,0,200,177,209
20001 DATA136,145,209,200,177,243,136,145,243,200,192,39,208,239,24,165,209
20002 DATA105,40,133,209,133,243,165,210,105,0,133,210,41,3,9,216,133,244,232
20003 DATA236,197,2,208,210,96,32,102,229,169,0,133,81,164,81,185,185,2,133
20004 DATA214,169,0,133,182,185,179,2,133,82,152,234,10,133,2,166,2,161,87
20005 DATA201,31,208,8,169,0,149,87,149,99,240,242,201,30,208,18,165,82,234
20006 DATA36,82,80,3,41,191,44,9,64,153,179,2,76,213,204,133,78,169,0,133,79
20007 DATA162,3,6,78,38,79,202,208,249,24,169,208,101,79,133,79,173,196,2,208
20008 DATA7,169,8,24,101,79,133,79,32,52,205,165,82,74,144,3,32,52,205,230
20009 DATA182,165,182,201,8,208,237,166,81,189,167,2,8,169,0,157,167,2,40,208
20010 DATA31,165,82,41,64,157,167,2,165,82,16,4,94,198,2,234,94,198,2,144,11
20011 DATA169,128,157,198,2,166,2,246,87,246,99,230,81,165,81,205,192,2,240
20012 DATA3,76,67,204,173,195,2,48,46,32,132,255,32,159,255,32,228,255,192
20013 DATA32,208,9,173,14,220,9,1,141,14,220,96,192,76,208,8,238,191,2,208
20014 DATA15,206,191,2,192,83,208,8,173,191,2,240,3,206,191,2,174,191,2,240
20015 DATA10,172,191,2,136,234,208,252,202,208,246,32,0,204,76,60,204,32,108
20016 DATA229,32,36,234,164,81,173,14,220,41,254,141,14,220,165,1,41,251,133
20017 DATA1,165,82,48,57,185,198,2,164,182,49,78,240,6,165,82,41,30,74,44,169
20018 DATA10,72,165,1,9,4,133,1,104,164,81,25,173,2,72,41,63,201,26,208,6,190
20019 DATA173,2,76,126,205,166,2,161,99,170,104,160,39,32,30,234,230,214,96
20020 DATA169,2,133,83,169,0,133,80,164,182,166,81,189,198,2,74,29,198,2,49
20021 DATA78,133,189,189,198,2,197,83,144,37,240,6,70,189,74,76,161,205,165
20022 DATA189,5,80,133,80,165,83,201,8,240,9,169,8,133,83,230,182,76,143,205
20023 DATA165,80,24,105,10,76,95,205,197,83,240,223,6,189,10,76,202,205,234
20024 DATA169,0,160,192,170,149,87,232,148,87,232,200,224,24,208,245,76,60
20025 DATA204,0,169,0,133,78,169,192,133,79,162,12,169,32,160,0,145,78,200
20026 DATA208,251,230,79,202,208,244,96,169,9,133,214,165,78,133,80,169,192
20027 DATA133,79,169,198,133,81,32,108,229,165,209,24,105,6,133,209,32,36,234
20028 DATA160,29,177,78,145,209,177,80,145,243,136,16,245,230,79,230,81,230
20029 DATA214,165,79,56,237,192,2,201,192,208,214,96,162,33,181,78,157,220,2
20030 DATA202,16,248,32,213,205,162,33,189,220,2,149,78,202,16,248,96,0,40,3,41
20031 DATA32,14,9,12,19,32,18,15,4,5,32,49,57,56,52,32
```

Caricatore HEX

Lunghezza del programma: 1000 byte circa. Linguaggio di programmazione: BASIC. Accessori: registratore o drive.

Molti programmi in questo fascicolo sono presentati sotto forma di listati esadecimali, in un formato adeguato a quello del monitor C64 da 40 caratteri. Per caricare questi programmi nel computer e poi memorizzarli su disco o cassetta, si può utilizzare questo breve programma "Caricatore HEX". Esso lavora esattamente nel formato utilizzato in questo fascicolo, cioè indirizzi a quattro cifre (output del computer) - otto byte dei dati (separati da virgola) - somma di controllo di quattro cifre.
L'utilizzazione del programma è molto semplice; se la somma introdotta e quella calcolata non corrispondono, il programma richiede di ripetere l'input della linea. Se non si ha tempo o voglia di ribattere il listato esadecimale, si può interrompere il lavoro, introducendo **FFFF** come somma di controllo. Il programma fornisce l'ultimo indirizzo memorizzato, che alla ripresa dell'operazione diventa l'indirizzo iniziale. Infine il codice generato fino a quel momento viene memorizzato a scelta su disco o su cassetta.
Per riunire le singole parti, si procede in questo modo:

LOAD "Partel",8,1
LOAD "Parte2",8,1 ... e così via
NEW
LOAD "CARICATORE HEX",8
SA = indirizzo iniziale dell'intero programma
EE = indirizzo finale
GOTO 440

Se un programma si introduce senza interruzioni, il tutto procede automaticamente. In ogni caso però, prima del lancio del "Caricatore HEX", alla linea 640 al posto di "FILENAME" bisogna introdurre il nome del programma corrente.
Tutti i programmi in codice macchina generati con questo sistema vengono caricati con LOAD "...",8,1, come di consueto.

Ulrich Kern

```
100 REM CARICATORE HEX
110 REM PER  CHIP-SPECIAL
120 REM DI  ULRICH KERN
130 REM -----------------
140 REM
150 PRINT"■";
160 PRINT"CARICATORE HEX"
170 PRINT"--------------"
180 PRINT:PRINT
190 DIMH$(15)
200 FORI=0TO15:READH$(I):NEXT
210 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
220 INPUT"IND INIZIALE (DECIMALE)";SA
225 S1=INT(SA/256):S2=SA-256*S1
226 IF SA>PEEK(45)+256*PEEK(46)THENPOKE55,S2:POKE56,S1
230 INPUT"IND FINALE+1  (DECIMALE)";EE
240 PRINT:PRINT
250 PRINT"FORMATO:"
260 PRINT"0000:00,00,00,00,00,00,00,00,0000":PRINT
270 S1=INT(SA/8)*8
280 S2=INT((EE+7)/8)*8
290 FORD=S1TOS2-1STEP8
300 GOSUB540:PRINTHE$;":";
310 OPEN1,0
320 INPUT#1,B$(0),B$(1),B$(2),B$(3),B$(4),B$(5),B$(6),B$(7)
    ,SS$:CLOSE1:PRINT
330 FE=0:FORI=0TO7:B$=B$(I):GOSUB500:FE=FE+B
340 IF(D+I)=SA)AND(D+I<EE)THENPOKED+I,BAND255
350 NEXTI
360 IFB<00RB>255THENFE=-1
370 B$=LEFT$(SS$,2):GOSUB500:PS=256*B:B$=RIGHT$(SS$,2):
    GOSUB500:PS=PS+B
380 IF PS<>65535 THEN 410
390 PRINT"INTERRUZIONE!"
400 PRINT"ULTIMO INDIRIZZO MEMORIZZATO:";D-1:EE=D:GOTO430
410 IFPS<>FETHENPRINT"ERRORE DI INPUT!":GOTO300
420 NEXTD
430 PRINT:PRINT
440 PRINT"MEMORIZZAZIONE"
450 PRINT
460 PRINT"F=FLOPPY  C=CASSETTE ?"
470 GETX$:IFX$<>"F"ANDX$<>"C"THEN470
480 POKE704,1:IFX$="F"THENPOKE704,8
490 GOTO 570
```

```
500 X$=LEFT$(B$,1):GOSUB510:B=X*16:X$=RIGHT$(B$,1):
    GOSUB510:B=B+X:RETURN
510 IFB$=""THENX=-1:RETURN
530 X=(ASC(X$)AND15)-9*(X$)"@"):RETURN
540 DH=INT(D/256):DL=D-DH*256
550 DD=DH:GOSUB560:HE$=X$:DD=DL:GOSUB560:HE$=HE$+X$:RETURN
560 HH=INT(DD/16):HL=DDAND15:X$=H$(HH)+H$(HL):RETURN
570 REM MEMORIZZAZIONE
580 SH=INT(SA/256):SL=SA-256*SH
590 EE=EE+2:EH=INT(EE/256):EL=EE-256*EH
600 POKE 43,SL:POKE44,SH
610 POKE 705,PEEK(45):POKE 706,PEEK(46)
620 POKE 707,EL:POKE708,EH
630 POKE 45,PEEK(707):POKE 46,PEEK(708)
640 SAVE"FILENAME",PEEK(704)
650 POKE 43,1:POKE 44,8 : POKE 45,PEEK(705):POKE 46,PEEK(706)
```

RAT-TAPE

Lunghezza del programma: 7200 byte. Linguaggio di programmazione: linguaggio macchina. Accessori: registratore.

Questo programma permette di memorizzare dati e programmi su cassetta ad una velocità sei volte maggiore del normale. I programmi memorizzati possono venire caricati con il comando LOAD, senza bisogno di un caricatore speciale. La struttura di RAT-TAPE prevede inoltre un avvio automatico dopo il caricamento mediante RUN.
Tutti i parametri utilizzati da RAT-TAPE possono essere modificati; l'unico inconveniente è che la sicurezza dei dati non è garantita al cento per cento come con il formato della versione originale, dal momento che la velocità di trasmissione è molto più elevata e che si rinuncia al SECOND PASS. La maggior frequenza si riflette soprattutto sulla precisione della regolazione della testina in fase di lettura e scrittura. Nel caso di lettura da un registratore estraneo, si possono presentare dei problemi. Inoltre il nastro magnetico è più sollecitato dal momento che i dropout lasciano dei "buchi" che non si possono correggere. Per questi motivi vi consigliamo di avere una copia del vostro programma memorizzata nel modo convenzionale, con SAVE.

MEMORIZZARE IN NU

Per memorizzare un programma dovete caricare RAT-TAPE con LOAD "RAT TAPE 51400",8,1 e dare un NEW. Adesso potete caricare il programma che desiderate memorizzare e lanciare RAT-TAPE con SYS 51400. La sequenza di start indicata è in realtà un segmento in codice macchina che conclude il processo di caricamento. La struttura è tale che dapprima viene preso dallo stack il bit di parità, confrontato col valore memorizzato in $A8 ed eventualmente viene visualizzato un messaggio "?LOAD ERROR". Se il processo di caricamento riesce, il puntatore BASIC di fine viene settato alla fine del programma e il puntatore CHRGET all'inizio del programma. Viene eseguito un CLR, è settato il flag di programma e si salta al loop dell'interprete, che equivale al comando RUN.
Normalmente non si deve cambiare o introdurre niente oltre al nome del file. Infine per memorizzare il programma, battete semplicemente il tasto F1; per tornare al BASIC, premete contemporaneamente i tasti RUN/STOP e RESTORE.

Ulrich Roller

```
RAT TAPE 51400
PER CHIP SPECIAL
DI ULRICH ROLLER

INDIRIZZO INIZIALE : 51400
INDIRIZZO FINALE   : 53248

C8C8:A9,93,20,D2,FF,A6,2B,A5,04A3
C8D0:2C,86,C1,85,C2,A6,2D,A5,0432
C8D8:2E,86,AE,85,AF,A2,C0,BD,04B5
C8E0:3F,CF,9D,3B,03,CA,D0,F7,047A
C8E8:A9,01,8D,86,02,A9,80,8D,0375
C8F0:91,02,A9,17,8D,18,D0,A2,036A
C8F8:04,8A,09,10,9D,F8,07,CA,030D
C900:10,F7,A2,00,BD,00,CE,9D,03D1
C908:00,04,A9,06,9D,00,D8,E0,0308
C910:40,B0,0B,BD,00,CF,9D,00,0324
C918:05,A9,06,9D,00,D9,E8,D0,03E2
C920:E3,A9,1F,A2,09,BD,E0,CD,04C0
C928:9D,00,D0,CA,10,F7,A9,01,03E8
C930:8D,27,D0,8D,28,D0,8D,29,03BF
C938:D0,A9,05,8D,2A,D0,8D,2B,03BD
C940:D0,A9,1F,8D,1D,D0,8D,15,03B4
C948:D0,A9,00,8D,17,D0,78,AE,0413
C950:9B,CD,AD,9C,CD,8E,14,03,0423
C958:8D,15,03,A9,7F,8D,0D,DC,0343
C960:AE,0D,DC,8D,0D,DD,AE,0D,03C9
C968:DD,A9,01,8D,1A,D0,AD,11,03BC
C970:D0,29,7F,8D,11,D0,58,A2,03E0
C978:0B,A0,00,A9,00,20,55,CC,0295
C980:A2,00,BD,41,03,20,49,CD,02D9
C988:E6,D3,E8,E0,10,90,F3,A2,05B6
C990:0D,A0,00,A9,10,20,55,CC,02A7
C998:A5,C2,20,6B,CC,A5,C1,20,0444
C9A0:6B,CC,A2,0F,A0,00,A9,20,0351
C9A8:20,55,CC,A5,AF,20,6B,CC,03EC
C9B0:A5,AE,20,6B,CC,A2,11,A0,03FD
C9B8:00,A9,30,20,55,CC,E6,D3,03D3
C9C0:A0,00,20,84,CC,A2,12,A0,0364
C9C8:0D,20,20,CC,A0,08,20,84,0265
C9D0:CC,A2,13,A0,0D,20,20,CC,033A
C9D8:A0,10,20,84,CC,A2,14,A0,0376
C9E0:0D,20,20,CC,A0,18,20,84,0275
C9E8:CC,A2,0B,A0,0D,20,20,CC,0332
C9F0:A9,10,20,A0,CC,48,A2,0F,033E
C9F8:BD,00,02,9D,41,03,CA,10,027A
CA00:F7,68,C9,85,F0,21,C9,0D,0494
CA08:F0,02,C9,11,D0,DB,A2,0D,0426
CA10:A0,0D,20,F2,CB,86,C1,84,0455
CA18:C2,C9,91,F0,CC,C9,0D,F0,059E
CA20:02,C9,11,F0,04,C9,85,F0,040E
CA28:0D,A2,0F,A0,0D,20,F2,CB,0348
CA30:86,AE,84,AF,C9,85,F0,1F,04C4
CA38:C9,91,F0,D2,C9,11,F0,02,04E8
CA40:C9,0D,D0,E5,A2,17,86,B5,047F
CA48:A2,00,86,D3,A2,11,A9,00,0357
CA50:20,BC,CB,90,D4,C9,85,F0,0549
CA58:0B,A2,12,A9,08,20,BC,CB,0317
CA60:90,EA,C9,85,F0,1A,A2,13,0487
CA68:A9,10,20,BC,CB,90,EA,C9,04A3
CA70:85,F0,0D,A2,14,A9,18,20,0319
CA78:BC,CB,90,EA,C9,85,D0,F3,0612
CA80:A6,C1,A5,C2,8E,7C,03,8D,0468
CA88:7E,03,A6,AE,A5,AF,8E,C4,047B
CA90:03,8D,C8,03,A2,18,A0,00,02B5
CA98:20,20,CC,A2,00,8E,86,02,02C4
CAA0:BD,EB,F0,08,29,7F,20,49,03B1
CAA8:CD,E6,D3,E8,28,10,F1,20,04B7
CAB0:2E,F8,F0,0B,20,E1,FF,D0,04F1
CAB8:F6,78,EA,EA,4C,66,FE,78,056A
CAC0:20,15,FD,20,A3,FD,20,18,032A
CAC8:E5,58,A5,AF,48,A5,AE,48,0474
CAD0:A5,C2,48,A5,C1,48,20,D7,0454
CAD8:F7,A9,69,85,AB,20,70,F8,04C1
CAE0:A2,7B,A9,03,86,FB,85,FC,04CB
CAE8:A2,FB,A9,00,86,C1,85,C2,04D4
CAF0:A2,FD,A9,00,86,AE,85,AF,04B0
CAF8:A9,14,85,AB,20,70,F8,68,03DD
CB00:85,AC,85,C1,68,85,AD,85,0496
CB08:C2,68,85,AE,68,85,AF,A9,04A2
CB10:36,85,01,A0,00,84,C0,AD,034D
CB18:11,D0,29,EF,8D,11,D0,CA,0431
CB20:D0,FD,88,D0,FA,78,A9,02,0542
CB28:85,AB,A0,00,A9,02,20,8D,0328
CB30:CB,A2,07,88,C0,09,D0,F4,0489
CB38:A2,05,C6,AB,D0,EE,98,20,048E
CB40:8D,CB,A2,07,88,D0,F7,A2,04F2
CB48:05,98,20,8D,CB,84,D7,A2,0412
CB50:07,EA,B1,AC,20,8D,CB,A2,0468
CB58:03,E6,AC,D0,04,CA,CA,E6,04E3
CB60:AD,A5,AC,C5,AE,A5,AD,E5,05A8
CB68:AF,90,E7,A5,D7,20,8D,CB,051A
CB70:A2,07,88,D0,F6,C8,84,C0,0503
CB78:58,A9,00,8D,A0,02,18,20,0268
CB80:93,FC,A9,37,85,01,85,C0,043A
CB88:4C,ED,C8,EA,EA,85,BD,45,055C
CB90:D7,85,D7,A9,08,85,A3,06,0412
CB98:BD,A5,01,29,F7,20,AF,CB,041D
CBA0:A2,11,EA,09,08,20,AF,CB,0348
CBA8:A2,0E,C6,A3,D0,E9,60,CA,04FC
CBB0:D0,FD,90,05,A2,0B,CA,D0,04A9
CBB8:FD,85,01,60,85,FB,A0,0D,0410
CBC0:A5,D3,20,20,CC,85,D3,20,03FC
CBC8:A6,CC,48,A2,00,A4,FB,20,041B
CBD0:09,CC,99,DC,03,E8,C8,E0,04DD
CBD8:17,90,F4,68,C9,85,F0,11,0452
CBE0:C9,91,18,F0,0C,C9,11,F0,0438
CBE8:08,C9,0D,D0,DA,A9,00,85,03B6
CBF0:D3,60,A0,0D,20,20,CC,A9,0395
CBF8:04,20,A0,CC,48,A2,00,20,029A
CC00:09,CC,A8,20,09,CC,AA,68,0384
CC08:60,BD,00,02,20,93,CC,0A,02A8
CC10:0A,0A,0A,85,FF,BD,01,02,0262
CC18:20,93,CC,05,FF,E8,E8,60,04B3
CC20:48,A9,00,85,D3,8A,A2,00,0375
CC28:85,D1,86,D2,0A,0A,18,65,033F
```

```
CC30:D1,85,D1,A2,03,06,D1,26,03C9
CC38:D2,CA,D0,F9,18,98,65,D1,054B
CC40:85,D1,90,02,E6,D2,85,F3,0518
CC48:18,A5,D2,69,04,85,D2,69,03BC
CC50:D4,85,F4,68,60,20,20,CC,0421
CC58:A2,00,86,D3,AA,BD,A0,CD,04CF
CC60:F0,08,20,49,CD,E6,D3,E8,04CF
CC68:D0,F3,60,48,4A,4A,4A,4A,0393
CC70:20,74,CC,68,29,0F,09,30,0239
CC78:C9,3A,90,02,69,06,20,49,026D
CC80:CD,E6,D3,60,A2,08,B9,DC,0525
CC88:03,20,6B,CC,E6,D3,C8,CA,04A5
CC90:D0,F4,60,C9,3A,08,29,0F,0367
CC98:28,90,02,69,08,60,00,00,018B
CCA0:85,B5,A0,00,84,D3,A4,D3,04A8
CCA8:B1,D1,85,CE,A9,01,85,CD,04D1
CCB0:A9,00,85,CF,85,CC,20,E4,0452
CCB8:FF,F0,FB,A0,01,84,CC,20,04FB
CCC0:49,CD,A4,D3,90,0A,48,A5,0414
CCC8:CE,91,D1,68,C9,1D,D0,0A,0458
CCD0:E6,D3,A4,D3,C4,B5,90,CE,0607
CCD8:B0,04,C9,9D,D0,0A,C6,D3,048D
CCE0:A4,D3,C0,FF,90,C0,B0,E8,061E
CCE8:C9,14,D0,17,A4,D3,F0,B6,04E1
CCF0:B1,D1,88,91,D1,C8,C8,C4,05C0
CCF8:B5,90,F5,88,EA,A9,20,91,0506
CD00:D1,B0,DB,C9,94,D0,1F,A4,054C
CD08:B5,88,B1,D1,C9,20,D0,13,048B
CD10:88,C4,D3,90,09,B1,D1,C8,0502
CD18:91,D1,88,EA,10,F2,C8,A9,0547
CD20:20,91,D1,4C,A6,CC,48,A4,042C
CD28:B5,88,30,18,B1,D1,85,FF,048B
CD30:29,3F,06,FF,24,FF,10,02,02A2
CD38:09,80,70,02,09,40,99,00,01DD
CD40:02,18,90,E5,68,60,00,00,0257
CD48:00,48,85,FF,29,3F,06,FF,0339
CD50:24,FF,30,02,50,13,90,02,024A
CD58:09,40,84,FF,A4,D3,91,D1,04A5
CD60:AD,86,02,91,F3,A4,FF,18,0474
CD68:24,38,68,60,A2,01,CA,D0,0361
CD70:FD,A6,02,E8,E8,E0,0C,90,04F1
CD78:02,A2,00,BD,F0,CD,8D,20,03CB
CD80:D0,8D,21,D0,BD,F1,CD,8D,0556
CD88:12,D0,86,02,A9,01,8D,19,02BA
CD90:D0,8A,D0,03,4C,31,EA,4C,03E0
CD98:81,EA,2C,6C,CD,2C,00,00,02FC
CDA0:C6,49,4C,45,4E,41,4D,45,02C1
CDA8:20,20,20,3A,20,00,00,00,00BA
CDB0:D3,54,41,52,54,20,C1,44,0333
CDB8:52,2E,20,3A,20,00,00,00,00FA
CDC0:C5,4E,44,20,C1,44,52,45,0313
CDC8:53,53,20,3A,20,00,FF,FF,031E
CDD0:D3,54,41,52,54,20,D3,45,0346
CDD8:51,2E,20,3A,00,00,FF,FF,02D7
CDE0:3F,48,6F,48,9F,48,CF,48,033C
CDE8:FF,48,FF,FF,00,00,FF,FF,0543
CDF0:00,32,06,72,01,7A,0B,F0,0220
CDF8:0E,FB,00,FF,FF,FF,FF,FF,0604
CE00:00,00,00,00,00,00,00,00,0000
CE08:00,00,7F,F0,00,FF,F8,01,0367
CE10:FF,FC,01,E0,7C,03,C0,3C,0457
CE18:03,C0,3C,07,80,3C,07,C0,0289
CE20:78,0F,FF,F8,0F,FF,F0,1F,049B
CE28:FF,C0,1E,03,C0,3C,03,C0,039F
CE30:3C,01,E0,FE,01,F0,FE,01,040B
CE38:F8,FE,01,FC,00,00,00,00,02F3
CE40:00,00,00,00,00,00,00,3F,003F
CE48:E0,00,FF,F0,01,FF,F8,03,04CA
CE50:E0,F8,03,C0,78,07,80,78,0412
CE58:07,80,78,0F,C0,F8,0F,FF,03D4
CE60:F8,1F,FF,F8,1F,FF,F8,3E,0562
CE68:00,F8,3C,00,78,78,00,78,029C
CE70:78,00,78,F8,00,FC,F8,00,03DC
CE78:FC,00,00,00,00,00,00,00,00FC
CE80:00,00,00,00,00,00,1F,FF,011E
CE88:FC,3F,FF,FC,7F,FF,FC,78,0628
CE90:FE,1C,00,7C,00,00,F8,00,028E
CE98:00,F8,00,01,F0,00,01,F0,02DA
CEA0:00,03,E0,00,03,E0,00,07,01CD
CEA8:C0,00,07,C0,00,0F,80,00,0216
CEB0:1F,80,00,3F,80,00,3F,C0,025D
CEB8:00,00,00,00,00,00,00,00,0000
CEC0:00,00,00,00,00,00,00,00,0000
CEC8:00,00,00,00,00,00,00,00,0000
CED0:00,00,0F,C0,E0,1F,E3,F8,03A9
CED8:1C,67,1C,1C,07,1C,1F,87,0184
CEE0:1C,0F,C7,1C,00,E7,1C,18,0229
CEE8:E7,1C,1F,E3,F8,0F,C1,E0,04AD
CEF0:00,00,00,00,00,00,00,00,0000
CEF8:00,00,00,00,00,00,00,00,0000
CF00:00,00,00,00,00,00,07,C0,00C7
CF08:E0,0F,E0,E0,1F,F0,E0,1C,04BA
CF10:70,E0,1C,00,E0,1C,07,FC,036B
CF18:7F,07,FC,7F,07,FC,7F,00,0383
CF20:E0,1C,00,E0,1C,00,E6,1C,02FA
CF28:00,FE,1C,00,FE,1C,00,7C,02B0
CF30:00,00,00,00,00,00,00,00,0000
CF38:00,00,00,00,00,00,00,00,0000
CF40:03,24,03,26,03,2E,2E,2E,00DD
CF48:2E,2E,2E,2E,2E,2E,2E,2E,0170
CF50:2E,2E,2E,2E,2E,A9,10,2C,01CB
CF58:0D,DC,F0,FB,AD,0D,DD,8E,04F9
CF60:07,DD,48,A9,19,8D,0F,DD,0367
CF68:68,4A,4A,60,A9,08,85,A3,0335
CF70:20,51,03,26,BD,EA,EA,EA,0415
CF78:C6,A3,D0,F4,A5,BD,60,A2,0591
CF80:01,A9,08,86,AC,85,AD,20,0336
CF88:17,F8,78,A2,0C,20,8D,F8,03DA
CF90:78,A9,07,8D,06,DD,A2,01,033B
CF98:20,51,03,26,BD,A5,BD,C9,0382
CFA0:02,D0,F5,A0,09,20,68,03,02FB
CFA8:C9,02,F0,F9,C4,BD,D0,E8,05ED
CFB0:20,68,03,88,D0,F6,20,68,0361
CFB8:03,91,AC,45,A8,85,A8,E6,0440
CFC0:AC,D0,02,E6,AD,A5,AC,C9,052B
CFC8:7E,A5,AD,E9,0F,90,E7,20,045F
```

```
CFDØ:68,Ø3,48,2Ø,93,FC,A2,57,Ø35B
CFD8:A9,F1,8E,24,Ø3,8D,25,Ø3,Ø3Ø4
CFEØ:68,C5,A8,FØ,Ø3,4C,9C,E1,Ø491
CFE8:A6,AC,A5,AD,86,2D,85,2E,Ø4ØA
CFFØ:2Ø,33,A5,2Ø,59,A6,A9,ØØ,Ø2CØ
CFF8:2Ø,9Ø,FF,4C,AE,A7,ØØ,ØØ,Ø35Ø
DØØØ:ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØØØ
```

Screensave

Lunghezza del programma: circa 3500 byte. Linguaggio di programmazione: Assembler. Accessori: registratore, drive.

Questo programma è stato concepito per memorizzare e caricare maschere video; nel modo grafico può servire anche per memorizzare i REM-colore. Dopo l'avvio con RUN, il programma li memorizza nell'area da $C000. L'ultima maschera video utilizzata è in $CC00. "Screensave" modifica il vettore RESET e mette a disposizione le seguenti variabili:

VARIABILI DI TIPO STRINGA

○ $,s
memorizza il contenuto del video sotto il nome definito nella stringa, che non deve superare i 16 caratteri di lunghezza.

○ $,l
carica una videata memorizzata sotto il nome contenuto nella stringa. Se questa videata non si trova, il programma visualizza l'ultima videata elaborata.

Avvertenza: se queste due stringhe non vengono definite (hanno lunghezza uguale a zero) o contengono più di 16 caratteri, viene visualizzato il messaggio: "?bad filename error".

○ $
lista le prime 22 linee della directory ed aspetta che venga battuto un tasto; poi continua il listato.

○ $,c
cancella il contenuto del video memorizzato temporaneamente.

○ $,r
richiama l'ultima videata elaborata.

POSSIBILITÀ DI MODIFICA

○ Evitare l'output di testo:
POKE 49153,29

○ Modificare il colore dello sfondo (solo per l'output di testi):
POKE 49161,(240 + numero del colore)

○ Modificare il colore dei caratteri (solo per l'output di testi):
POKE 49223,(codice del colore)

○ Modificare il tabulatore dell'inizio colonna della directory:
POKE 49517,(tab) (almeno 1!)

○ Modificare il numero delle linee di directory da visualizzare:
POKE 49474,(linee)

○ Evitare che il contenuto del video venga riprodotto in un unico colore:
POKE 49677,76:POKE 49678,50:POKE49679,194

Tutte le modifiche restano valide fino allo spegnimento del calcolatore (anche se si dà un RESET).

ESEMPIO DI PROGRAMMA

Linea 5: setta primo piano e sfondo blu, caratteri bianchi
Linee 50-60: visualizzano la prima maschera video
Linea 70: memorizza la prima maschera sotto il nome "screendemo 1" (le virgolette fanno in modo che tutti i programmi con questo nome vengano riscritti)
Linee 105-120: visualizzano la seconda maschera
Linea 130: memorizza la seconda maschera sotto il nome "screendemo 2"
Linea 140-160: caricano la prima maschera
Linee 170-175: posizionano il cursore, aspettano un input
Linee 200-220: caricano la seconda maschera
Linee 230-235: posizionano il cursore, aspettano un input
Linee 240-270: listano la directory

Christoph Rilke

```
150 DATA2398,2386,2538,1931,1240,1324,1990,2418,2405,2378
    ,2262,2601,2436
160 DATA2666,2769,2790,2491,2454,2630,2822,3298,1953,2473
    ,1881,3033,2965
170 DATA3083,3498,3106,3203,2445,3025,3526,2942,2897,2081,645
180 DATA32,8,192,169,0,76,231,167,169,246,141,32,208,141
    ,33,208,160,0,185,-1
190 DATA71,192,240,6,32,22,231,200,208,245,169,117,160
    ,192,141,8,3,140,9,-1
200 DATA3,162,5,189,15,253,157,3,128,202,208,247,169,176
    ,160,192,141,0,128,-1
210 DATA140,1,128,169,94,160,254,141,2,128,140,3,128,96,5
    ,147,13,32,83,67,-1
220 DATA82,69,69,78,83,65,86,69,13,13,40,67,41,32,49,57,56
    ,52,32,66,89,32,-1
230 DATA67,72,82,46,32,82,73,76,75,69,13,13,0,0,0,0,0,0
    ,32,115,0,201,36,240,-1
240 DATA6,32,121,0,76,231,167,32,115,0,208,3,76,215,192
    ,201,44,208,31,32,-1
250 DATA115,0,201,67,208,3,76,118,193,201,76,208,3,76
    ,189,193,201,82,208,-1
260 DATA3,76,223,193,201,83,208,3,76,65,194,76,8,175,162
    ,255,142,22,208,32,-1
270 DATA80,253,32,21,253,32,163,253,32,91,255,88,32,83
    ,228,32,191,227,32,-1
280 DATA8,192,32,68,166,162,251,154,32,96,166,76,134
    ,227,32,68,229,169,0,-1
290 DATA141,117,193,32,106,193,169,36,133,251,169,251,133
    ,187,169,0,133,188,-1
300 DATA169,1,133,183,169,8,133,186,169,96,133,185,32
    ,213,243,165,186,32,-1
310 DATA180,255,165,185,32,150,255,169,0,133,144,160,3
    ,132,251,32,165,255,-1
320 DATA133,252,164,144,208,74,32,165,255,164,144,208
    ,67,164,251,136,208,-1
330 DATA233,166,252,32,205,189,169,32,32,210,255,32,165
    ,255,166,144,208,45,-1
340 DATA170,240,6,32,210,255,76,46,193,173,117,193,201
    ,22,208,14,32,62,241,-1
350 DATA240,251,169,0,141,117,193,169,147,208,2,169,13
    ,32,210,255,32,106,-1
```

```
360 DATA193,238,117,193,160,2,208,171,32,66,246,169,0,76
    ,231,167,169,32,160,-1
370 DATA6,32,210,255,136,208,250,96,0,169,0,160,204,133,250
    ,132,251,168,162,-1
380 DATA204,169,32,145,250,200,208,251,224,207,240,5
    ,232,230,251,208,242,-1
390 DATA76,228,167,32,115,0,201,44,240,2,56,96,32,115
    ,0,32,139,176,32,170,-1
400 DATA182,216,56,233,16,16,239,24,105,16,141,186,193,142
    ,187,193,140,188,-1
410 DATA193,24,96,0,0,0,32,148,193,144,3,76,161,194
    ,162,8,160,1,32,0,254,-1
420 DATA173,186,193,240,241,174,187,193,172,188,193,32
    ,249,253,169,0,32,158,-1
430 DATA244,160,8,185,247,7,153,247,207,136,208,247
    ,169,0,160,204,133,250,-1
440 DATA132,251,160,4,133,252,132,253,168,162,204,177
    ,250,145,252,200,208,-1
450 DATA249,224,207,240,7,232,230,251,230,253,208
    ,238,169,216,160,0,133,251,-1
460 DATA132,250,170,173,134,2,145,250,200,208,251,224
    ,218,208,7,169,232,133,-1
470 DATA250,232,208,237,224,219,240,5,230,251,232,208
    ,231,32,121,0,201,82,-1
480 DATA208,3,76,228,167,169,0,76,231,167,32,148,193,144
    ,3,76,161,194,169,-1
490 DATA0,160,4,133,250,132,251,160,204,133,252,132,253
    ,168,162,204,177,250,-1
500 DATA145,252,200,208,249,224,207,240,7,232,230,251,230
    ,253,208,238,152,-1
510 DATA160,204,133,251,132,252,162,8,32,0,254,173,186,193
    ,240,201,174,187,-1
520 DATA193,172,188,193,32,249,253,169,251,162,0,160
    ,208,32,221,245,169,0,-1
530 DATA76,231,167,66,65,68,32,70,73,76,69,78,65,77
    ,197,0,169,148,160,194,-1
540 DATA133,34,132,35,76,71,164,-1
550 DIMS(36):FORX=0TO36:READS(X):NEXT:PRINT"■"
560 FORX=49152TO49835
570 PRINT"■LOCAZIONE :"X:PRINT"LINEA          :"PEEK(63)
580 READD:IFD<>-1THEN610
590 IFS=S(Z)THENS=0:Z=Z+1:X=X-1:GOTO640
600 GOTO660
610 IFD<0ORD>255THEN660
620 POKEX,D
630 S=S+D
640 NEXT
650 GOTO690
660 PRINT"■■■■■■■ERRORE NEI DATA!■"
670 PRINT"■■L、"Z+3"□□";
680 POKE631,13:POKE198,1:END
690 PRINT"■■■■M■EMORIZZARE O    ■L■ANCIARE?"
700 GETI$:IFI$=""THEN700
710 IFI$="M"THENSAVE"SCREENSAVE",8:GOTO690
720 IFI$="L"THENSYS49152:NEW
730 GOTO700
```

```
5 POKE53280,6:POKE53281,6:PRINT"■"
10 PRINT"■PROGRAMMA ESEMPIO PER'SCREENSAVE'"
20 PRINT"■■■GENERAZIONE E MEMORIZZAZIONE          PRIMA MASCHERA"
30 PRINT"■■■PREMERE UN TASTO!"
40 GETI$:IFI$=""THEN40
50 PRINT"■                    INPUT DATI"
60 PRINT"■■■■■■INTRODURRE QUI:"
70 F1$="@:SCREENDEMO 1":$,S,F1$
80 PRINT"■GENERAZIONE E MEMORIZZAZIONE SECONDA     MASCHERA"
90 PRINT"■■■PREMERE UN TASTO!"
100 GETI$:IFI$=""THEN100
105 PRINT"■"
110 FORX=1024TO2023:POKEX,160:NEXT
120 POKE1523,32
130 F2$="@:SCREENDEMO 2":$,S,F2$
140 PRINT"■CARICAMENTO PRIMA MASCHERA"
150 F1$=RIGHT$(F1$,12)
160 $,L,F1$
170 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
175 OPEN1,0:INPUT#1,D$:CLOSE1:PRINTCHR$(13)
200 PRINT"■CARICAMENTO SECONDA MASCHERA"
210 F2$=RIGHT$(F2$,12)
220 $,L,F2$
230 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
235 OPEN1,0:INPUT#1,E$:CLOSE1:PRINTCHR$(13)
240 PRINT"■DIRECTORY"
250 PRINT"PREMETE UN TASTO!"
260 GETI$:IFI$=""THEN260
270 $:END
```

Dottore dei dischetti

Lunghezza del programma: 3500 byte. Linguaggio di programmazione: BASIC. Accessori: drive per dischetti.

Avete mai avuto dei problemi nella gestione dei dischetti?
Per esempio se manca la corrente quando un file di dati, a cui magari avete lavorato per ore, è non ancora chiuso. Oppure quando scrivete un programma, che lavora in handshake col dischetto; cercate di accedervi con un comando di accesso diretto e scoprite che la directory è danneggiata irreparabilmente.
Le disavventure nel mondo dei dischetti possono essere di vario genere.
Se però siete tra i fortunati utenti del programma "Dottore dei dischetti", sarete in grado di risolvere questi problemi. Anche se non siete un esperto, vi stupirete di quante cose potete fare col "Dottore dei dischetti".
L'autore si è deciso a scrivere questo programma dopo spiacevoli esperienze con i dischetti, proprio per avere uno strumento di lavoro adeguato. Si è sforzato di scrivere un programma confortevole e comprensibile anche per gli altri utenti.
Nonostante tutte le misure di sicurezza che il programma contiene, bisogna usarlo in modo consapevole. Chi non conosce bene il programma deve rinunciare al menù "6", la routine per inviare i dati al drive e al dischetto, e per usare tranquillamente il menù "7" non basta avere letto il manuale. Le conoscenze fondamenti sono le seguenti:
1. Elaborazione della directory: i singoli file vengono prodotti e possono essere sottoposti a editing.
2. Disposizione dei blocchi (RAM): la matrice prodotta mostra la disposizione dei blocchi e ne permette l'editing.
3. Nome del dischetto: la denominazione del dischetto diventa visibile e si può sottoporre ad editing. Inoltre si può proteggere tutto il disco con l'attivazione del menù "7" che non permette alcuna variazione del suo contenuto e può essere rimossa soltanto con un ID formattato.

4. Ordinamento della directory: questo menù richiede i numeri sotto cui sono stati listati i file del menù "1". Una volta che i numeri sono stati ordinati, non si può introdurre un altro valore, perciò l'input termina con una lettera o con uno zero. L'ordinamento precedente viene annullato, tutti i file non prodotti vengono eliminati e l'ordinamento tiene conto dei tipi dei file. Il ripristinino dell'ordinamento vecchio si deve decidere già al momento del primo input; il nuovo ordinamento viene convalidato all'esecuzione del menù "7".

5. Listato dei blocchi dei file: dopo l'input del numero del file come dal menù "1", vengono prodotti i blocchi appartenenti a quel file. Nel caso di file relativi, non si tiene conto del side-sector block.

6. Invio di modifiche al dischetto: alla sua chiamata viene presentata un'opzione per selezionare la parte di lavoro da inviare al dischetto.

7. Finestra dei blocchi: un determinato blocco del dischetto viene rappresentato sotto forma di matrice per essere sottoposto ad editing. Il cursore si muove nel modo consueto, passa sempre al simbolo grafico successivo e si ferma sul "low byte". I valori esadecimali introdotti vengono shiftati verso sinistra.

8. Avvertenze: intervenendo alle linee 1340-1430 l'utente può formulare delle avvertenze personalizzate.

9. Fine: tramite menù il programma si può terminare o rilanciare. Nel caso in cui sotto il menù "7" siano stati caricati più di 15 blocchi, si consiglia un nuovo lancio.

Gerwin Becker

```
100 REM**********************************
110 REM*  (C) GERWIN BECKER            *
120 REM*  (C) TECNICHE NUOVE           *
130 REM*        VIA MOSCOVA 46/9A-MILANO*
140 REM**********************************
150 OPEN11,2,6,CHR$(8)
160 J%=1:K%=1:L%=0:N%=1:M%=0:O%=0:P%=0:Q%=1:R%=1:S%=0:T%=18
170 U%=0:V%=0:W%=0:X%=20:Y%=15:Z%=15
180 T=53272:POKET,23:POKET+9,0:POKE649,1:POKET+8,2:
    REM  POKE788,52:POKE775,213
190 PRINTCHR$(147)"■■■■■■■■■■■■■■            ■ LETTURA -ISCO  ■":
    GOSUB780
200 DIMA%(X%,Y%,Z%),C%(36,21),G%(144)
210 ONJ%GOSUB340,1620,2200,2680,2890,4030,3560,3110,4130
220 FORA=U%TOX%:ONK%GOSUB490,550,650,3680,3990
230 FORB=V%TOY%:ONL%GOSUB100,2290,560,3700
240 FORC=W%TOZ%:ONN%GOSUB670,740,3280,4150
250 ONM%GOSUB790,730,4150,2340,3310,3730
260 ONO%GOSUB930,970,1040,1090,1190,1270,1370,1420,1550,1590
270 NEXTC:ONP%GOSUB1690,2360,2730,630,3760
280 NEXTB:ONQ%GOSUB820,870,2360,2360,3190,3370,780,3600,710,3780
290 NEXTA:X%=0:Y%=0:Z%=0:ONR%GOTO210,2610,2600,3800,2390
300 CLOSE1:CLOSE2:CLOSE3:END
310 END
320 :
330 REM ---- DISPLAY ----
340 PRINTCHR$(147)+"■■ ,N ¬ROGRAMMA DI   |‾_o\/ |‾-‾_          ■"
350 PRINT"  (-)  ■■   ‾\♥┘ - ┬-   ■  1985
360 PRINT"■---------------------------------------
370 PRINT"    1    ¬LABORAZIONE DIRECTORY
380 PRINT"■   2    -ISPOSIZIONE BLOCCHI (|♠\)
390 PRINT"■   3    /OME DEL DISCO
400 PRINT"■   4    ORDINAMENTO DELLA DIRECTORY
410 PRINT"■   5    LISTATO DEI BLOCCHI
420 PRINT"■   6    \NVIO \ODIFICHE AL -ISCO
430 PRINT"■   7    _INESTRA DEI BLOCCHI
440 PRINT"■   8    ♠VVERTENZE
450 PRINT"■   9    *** _ \ / ‾ ***
460 J%=0:RETURN
```

```
470 :
480 REM ---- LETTURA ERRORI  ----
490 GOSUB4310:A=0
500 :
510 REM ---- LETTURA DIRCTORY ----
520 OPEN1,8,15:H%=0
530 PRINT#1,"U:":FORD=0TO6↑4:NEXT:CLOSE1
540 OPEN1,8,15:OPEN2,8,2,"#":I%=0
550 C%(0,A)=64*T%+S%:C%(T%,S%)=A+1
560 K%=2:IFE%<ATHENE%=A
570 PRINT#1,"U1 2 0";T%;S%
580 INPUT#1,F%:IFF%>0ANDF%<30THENPRINTF%;F$(F%-20):GOTO2640
590 PRINT#1,"B-P 2";I%
600 RETURN
610 :
620 REM ---- STRINGA NEL BUFFER DEL DISCO ---
630 PRINT#2,E$;
640 REM ---- CANCELLA STRINGA  ----
650 E$="":RETURN
660 :
670 REM ---- LETTURA BYTE DA DISCO ----
680 GET#2,E$:A%(A,B,C)=ASC(E$+CHR$(0)):RETURN
690 :
700 REM ---- BUFFER SUL SETTORE ----
710 PRINT#1,"U2 2 0";T%,S%:GOTO580
720 :
730 REM ---- LETTURA BYTE ----
740 E$=E$+CHR$(A%(A,B,C)):IFC+B=0THENE$=CHR$(A%(A,B,C))
750 RETURN
760 :
770 REM ---- INPUT COMANDO  ----
780 WAIT203,64:WAIT203,63
790 A$="":GETA$:O%=VAL(A$):IFA$="@"THENGOSUB4250:GOTO790
800 RETURN
810 :
820 REM ---- BLOCCO SEQUENZIALE ---
830 T%=A%(A,0,0):S%=A%(A,0,1):M%=1
840 IFT%=0ORT%>35THENJ=0:K%=0:N%=0:M%=2:A=X%:U%=X%:Q%=0
    :IFH%=0THENH%=E%
850 RETURN
860 :
870 R%=2:RETURN
880 :
890 J%=1:K%=0:L%=0:N%=0:M%=2:P%=0:Q%=0:R%=1:U%=0:V%=0:W%=0:
    X%=1:Y%=1:Z%=1
900 GOTO210
910 :
920 REM ---- DIRECTORY ----
930 PRINTCHR$(147)"■■■■■■■■■■■■       ■ -IRECTORY ■
940 M%=0:O%=0:J%=2:RETURN
950 :
960 REM ---- BAM  ----
970 PRINTCHR$(147)+"■■■■■    LA SEGUENTE MATRÍCE DA'"
980 PRINT"■■    LA DISPOSIZIONE DEI BLOCCHI (|♠\)
990 PRINT"■■■■QUIVALE A:■  .  OCCUPATO   ■"
1000 PRINT"■■              ■  X  LIBERO     ■"
1005 FORT=1TO2500:NEXT
1010 M%=0:O%=0:J%=3:RETURN
1020 :
```

```
1030 REM ---- CARATTERE DEL DISCO ----
1040 PRINTCHR$(147)+"■          ■■  ─ARATTERE DEL DISCO  ■■"
1050 PRINT"■----------------------------------------■
1060 M%=0:O%=0:J%=4:RETURN
1070 :
1080 REM ---- ORDINAMENTO DELLA DIR ---
1090 PRINTCHR$(147)+"■QUANDO VI VIENE RICHIESTO, INTRODUCETE
1100 PRINT"■I NUMERI DEI FILE NELL'ORDINE DESIDERATO
1110 PRINT"■┐ FILE NON CHIAMATI VENGONO
1120 PRINT"■CANCELLATI DALLA ─┐_. ┐ BLOCCHI COSI'
1130 PRINT"■LIBERATI RESTANO INALTERATI.
1140 PRINT"■,NO '0' INIZIALE DISATTIVA
1150 PRINT"■L'OPERAZIONE, CIOE' CONCLUDE L'INPUT.■
1160 M%=0:O%=0:J%=5:RETURN
1170 :
1180 REM ---- LISTATO BLOCCHI ----
1190 PRINTCHR$(147)"■■VENGONO LISTATI I BLOCCHI DEI
     NUMERI DEIFILE INDICATI
1200 PRINT
1210 PRINT"■            LISTATO BLOCCHI
1220 PRINT"■----------------------------------------
1225 FORT=1TO1500:NEXT
1230 :
1240 M%=0:O%=0:J%=6:RETURN
1250 :
1260 REM ---- INVIO MODIFICHE ----
1270 PRINTCHR$(147)+"■♥CEGLI QUALI MODIFICHE DEVONO ESSERE
1280 PRINT"■INVIATE AL DISCO.
1290 PRINT"■■   A    ─IRECTORY
1300 PRINT"■   B    /OME DEL DISCO E ┐─
1310 PRINT"■   C    |♠\
1320 PRINT"■   D    DEF. |LOCCHI
1330 PRINT"■   E    |UTTO
1340 M%=0:O%=0:J%=7:RETURN
1350 :
1360 REM ---- FINESTRA ---
1370 PRINTCHR$(147)"■■■■   ■ ΓUTPUT ■ DI UN BLOCCO  DEF. LIBERO
1380 :
1390 M%=0:O%=0:J%=8:RETURN
1400 :
1410 REM ---- AVVERTENZE ---
1420 PRINTCHR$(147)"■ *  PER CONCLUDERE LA _OUTINE    "
1430 PRINT"■ # PER CORREGGERE (1,3)
1440 PRINT"■ @ PER L'OUTPUT SU VIDEO
1450 PRINT" ΓUTPUT  :X24(1200|D,NO ¬ARITY,1♥TOPBIT,          7BIT
     PER I DATI
1460 PRINT"          LINEE :90,4210-4240
1470 PRINT"■ LA PROTEZIONE TOTALE DEL DISCO NON VIEN E SEGNALATA";
1480 PRINT" E NON PUO' ESSERE ELIMINATA
1490 PRINT" LA PROTEZIONE IMPEDISCE DI MODIFICARE   IL DISCHETTO.
1500 PRINT" SOTTO LA ROUTINE 5 NEI FILE _¬L
1510 PRINT" ┐ BLOCCHI ♥IDE-♥ECTOR RESTANO NASCOSTI.
1520 GOSUB780
1530 M%=1:J%=1:RETURN
1540 :
1550 REM ---- FINE ----
1560 PRINTCHR$(147)"■■■■■                ■ ♠NCORA ? ■
1570 PRINT"■■■                             ■S■ / ■N■
1580 GOSUB780
```

```
1590 IFA$="S"THENRUN
1600 IFA$<>"N"THENRETURN
1610 M%=0:Q%=0:R%=0:RETURN
1620 :
1630 REM ---- CARATTERI DIR ----
1640 K%=3:L%=0:N%=2:M%=0:O%=0:P%=1:Q%=2:R%=1:U%=1:V%=0:W%=5
     :X%=H%:Y%=15:Z%=15
1650 PRINTCHR$(147)"█████████████":INPUT"/UM REGISTRAZIONE —IR";
     A$:X=VAL(A$)-1
1660 U%=1+X/8:V%=2*(XAND7)
1670 E$="":RETURN
1680 :
1690 IFINT(B/2)<B/2THEN1720
1700 W%=0:Z%=4:RETURN
1710 :
1720 IFASC(E$)<32THENB=Y%+1:RETURN
1730 G=A*8+B/2-7.5:PRINTCHR$(147)+"█ _EGISTRAZIONE DELLA —IR N.█"
     ;G;"█ █"
1740 PRINT"█----------------------------------------
1750 PRINT"██ —ENOMINAZIONE FILE : █";E$;"█":E$="   "
1760 I%=A%(A,B-1,2)AND7
1770 ONI%+1GOSUB2130,2140,2150,2160,2170
1780 PRINT"██ ●UESTO _ILE ";E$:J%=0
1790 PRINT"█ XENGONO OCCUPATI"+STR$(A%(A,B,14)+256*A%(A,B,15))
     +" |LOCCHI."
1800 PRINT"█ 1.|LOCCO: SULLA TRACCIA";A%(A,B-1,3);
     " ♥ETTORE";A%(A,B-1,4)
1810 IF(A%(A,B-1,2)AND4*16)<>0THENPRINT"██ ¬L _ILE E' PROTETTO.":J%=64
1820 IF(A%(A,B-1,2)AND8*16)=0THENPRINT"██ ¬L FILE NON E' STATO CHIUSO"
1830 W%=5:Z%=15:E$=""
1840 PRINT"----------------------------------------"
1850 PRINT"       * = \ENU      |     # =  —ORREZIONE"
1860 GOSUB780
1870 IFA$="*"THENB=15:A=E+2:N%=0:P%=0:RETURN
1880 IFA$<>"#"THENA$="":RETURN
1890 PRINT"----------------------------------------"
1900 PRINT"   ¬NTRODUCETE IL NOME DEL FILE.          ****
          ****
1910 PRINTCHR$(19):INPUT"█                           █████:█";A$
1920 FORD=5TO15:X=PEEK(D+507):IFX=0THENX=160
1930 A%(A,B-1,D)=X:NEXT
1940 FORD=0TO5:X=PEEK(D+523):IFX=0THENX=160
1950 A%(A,B,D)=X:NEXT
1960 PRINTCHR$(147)+"██ ¬L FILE █"+E$+"█ /.";G
1970 IFI%>0THENPRINT"█ VIENE CHIUSO (IL |IPO █—EL█ NO)."
1980 PRINT"█----------------------------------------"
1990 PRINT"███ —┐L=0 | ♥¯●=1 | ┐L|=2 | ♠_=3 | _┐L=4 ":E$=""
2000 INPUT"█ ¬NTRODUCETE LA CIFRA DI RICONOSCIMENTO";A$
2010 IF(A$<>""ANDVAL(A$)=0)ORVAL(A$)>4THEN2030
2020 IFE$>"/"ANDE$<"5"THENI%=VAL(E$)
2030 PRINT"██ ¯ROTEZIONE ♥OFTWARE █S█I/ █N█O"
2040 GOSUB780
2050 IFA$="S"THENJ%=64:PRINT"██ -- ¯ROTEZIONE ---
2060 IFA$="N"THENJ%=0
2070 INPUT"█ /UMERO PSEUDO BLOCCHI :";A$
2080 G=VAL(A$)
2090 IFA$<>""THENA%(A,B,14)=255ANDG:A%(A,B,15)=255AND(G/256)
2100 IFI%>0THENA%(A,B-1,2)=I%+J%+128
2110 RETURN
```

```
2120 :
2130 E$="E' STATO■ CANCELLATO ■ . LA DENOMINAZIONE DEL FILE E' ■■■■■■■
     ETED":RETURN
2140 E$="E' ■■■UENZIALE.":RETURN
2150 E$="E' UN ■■■■■■■■■RAMMA.":RETURN
2160 E$="E' UN FILE ■■■■■■■■.":RETURN
2170 E$="E' UN FILE■■■■■■■■ATIVO CON SEGNO"+STR$(A%(A,B,7))+". "
2175 +". "+CHR$(13)
2180 E$=E$+"■ L PRIMO BLOCCO ♥IDE-♥ECTOR E' SULLA TRACCIA"
     +STR$(A%(A,B,5))
2190 E$=E$+" E SUL SETTORE"+STR$(A%(A,B,6))+".":RETURN
2200 :
2210 REM ---- BAM ----
2220 K%=0:L%=2:N%=0:M%=4:O%=0:P%=2:Q%=0:R%=5:U%=0:V%=4:W%=0
     :X%=8:Y%=12:Z%=20
2230 PRINTCHR$(147)"                ■1■11111111■2■22222222■3■33333
2240 PRINT"     ■1■23456789■0■123456789■0■123456789■0■12345
2250 PRINT"  0■■1■■2■■3■■4■■5■■6■■7■■8■■9■■10■■1■■2■■3■■4■■
      5■■6■■7■■8■■9■■20
2260 PRINT" ■      * = \ENU                              ■
2270 RETURN
2280 :
2290 POKE49744,A%(0,A,B+1):POKE49745,A%(0,A,B+2):E$="□"
2300 POKE49746,A%(0,A,B+3):SYS49152
2310 V%=0:T%=A*4+B/4:
2320 Z%=((T%>17)*2+21+(T%>24)+(T%>30))-1:RETURN
2330 :
2340 E$=E$+CHR$(157)+CHR$(17)+CHR$(PEEK(50000+C)):RETURN
2350 :
2360 PRINTCHR$(19)+CHR$(17):POKE211,A*4+B/4+4:PRINTE$;:V%=0
     :E$="":B=B+3:RETURN
2370 :
2380 G=84:X=40
2390 G=GAND1023:Y=1024+G:POKEY,128ORPEEK(Y)
2400 FORD=0TO10:NEXT:A$=""
2410 GETA$:POKEY,127ANDPEEK(Y)
2420 IFA$=CHR$(64)THENGOSUB4250
2430 IFA$=CHR$(29)THENG=G+1
2440 IFA$=CHR$(157)THENG=G-1
2450 IFA$=CHR$(17)THENG=G+40
2460 IFA$=CHR$(145)THENG=G-40
2470 IFA$=CHR$(19)THENG=0
2480 IFA$=CHR$(42)THEN2610
2490 S%=INT(G/40)-2:IFINT(S%/21)<>0THEN2380
2500 T%=G-S%*40-84:IFINT(T%/35)<>0THEN2380
2510 IFA$<>"X"ANDA$<>"."THEN2390
2520 IFPEEK(Y)=ASC(A$)AND63THEN2390
2530 X=2↑(S%AND7)
2540 K%=A%(0,INT((T%+1)/4),4*((T%+1)AND3)+1+S%/8)AND(255ANDNOTX)
2550 X=-X*(A$="X")
2560 A%(0,INT((T%+1)/4),4*((T%+1)AND3)+1+S%/8)=K%ORX
2570 A%(0,INT(T%/4),4*(T%AND3))=A%(0,INT(T%/4),4*(T%AND3))
     +SGN(X)+(X=0)
2580 POKEY,ASC(A$)AND63:GOTO2390
2590 :
2600 GOSUB780
2610 GOSUB2640:GOTO210
2620 :
2630 GOSUB340
```

```
2640 I%=0:J%=1:K%=0:L%=0:M%=2:N%=0:O%=0:P%=0:Q%=0:R%=1:U%=1:V%=1:W%=1
2650 X%=1:Y%=3:Z%=5:E$="":RETURN
2660 :
2670 :
2680 :
2690 REM ---- CARATTERE DEL DISCO ---
2700 Q%=0
2710 K%=3:L%=0:N%=2:M%=0:O%=0:P%=3:R%=1:U%=0:V%=9:W%=0:X%=0
     :Y%=0:Z%=15:RETURN
2720 :
2730 PRINT"█                    ●UESTO DISCO HA "
2740 PRINT"██IL /OME                    :█"E$
2750 PRINT"█L'¬-                        :█"CHR$(A%(0,10,2));CHR$(A%(0,10,3))
2760 PRINT"█IL -ORMATO                  :█"CHR$(A%(0,10,5));CHR$(A%(0,10,6))
2770 PRINT"█----------------------------------------
2780 PRINT"█     # = -ORREZIONE
2790 GOSUB780:IFA$<>"#"THEN2880
2800 PRINTCHR$(19)"████████████████████████████████████";:INPUTA$
2810 FORD=0TO15:X=PEEK(D+512):IFX=0THENX=160
2820 A%(0,9,D)=X:NEXT
2830 INPUT"██████████████████████";A$
2840 FORD=0TO1:A%(0,10,D+2)=PEEK(D+512):NEXT
2850 PRINT"█  ¯ROTEZIONE TOTALE DEL DISCO █S█ / █N█"
2860 GOSUB780:O%=0:IFE$="S"THENA%(0,0,2)=77
2870 IFE$="N"THENA%(0,0,2)=65
2880 Q%=0:P%=0:R%=2:RETURN
2890 :
2900 REM ---- ORDINAMENTO DIR ---
2910 PRINT"█¬NPUT█>":G%(0)=1
2920 A$="":PRINT"-IVENTA IL -ILE N.█"G%(0);:INPUTA$:G=VAL(A$)
2930 IFG>H%*8THENPRINT"█";:GOTO2920
2940 G%(G%(0))=G:IFG%(0)=8*H%ORG=0THEN2630
2950 G%(0)=G%(0)+1:GOTO2920
2960 :
2970 REM ---- INPUT DEF. BLOCCHI ---
2980 A$="":PRINT"----------------------------------------"
2990 INPUT" ¬NTRODURRE LA TRACCIA :";A$
3000 IFA$=""THENRETURN
3010 T%=VAL(A$):IFT%<36THEN3040
3020 PRINT" █ -I SONO SOLO 35 |RACCE █"CHR$(157)CHR$(157)
3030 GOTO2990
3040 A$="":INPUT" ¬NTRODURRE IL SETTORE :";A$
3050 S%=((T%>17)*2+21+(T%>24)+(T%>30))
3060 IFS%>VAL(A$)THENS%=VAL(A$):RETURN
3070 PRINT"█         █♥U QUESTA TRACCIA ESISTONO█"
3080 PRINT"█      █SOLO█"S%"█ ♥ETTORI                 █▢▢▢▢▢"
3090 GOTO3040
3100 REM ---- CARATTERE FINESTRA ---
3110 J%=0:K%=2:L%=0:N%=1:M%=0:O%=0:P%=0:Q%=5:R%=2:S%=0:T%=18:U%=E%
3120 V%=0:W%=0:X%=U%+1:Y%=15:Z%=15
3130 GOSUB2980:IFA$="*"ORT%=0THEN2630
3140 IFC%(T%,S%)=0THENE%=E%+1:U%=E%:X%=E%+1:RETURN
3150 PRINTCHR$(147)"████ ●UESTO |LOCCO E' GIA' STATO CARICATO!
3160 PRINT"██ -EVE ESSERE RICARICATO ?  S / N":GOSUB780:
3170 IFA$="S"THENU%=C%(T%,S%)-1:RETURN
3180 U%=C%(T%,S%)-1
3190 K%=0:L%=0:N%=3:M%=5:O%=0:B=0:C=0
3200 IFA$="N"THEN3230
3210 PRINT"██ ┌UTPUT  S/N
```

```
3220 GOSUB780:IFA$="N"THEN2630
3230 PRINTCHR$(147)"     ■ ΓUTPUT |LOCCO |RACCIA"T%"■ ♥ETTORE "
     S%"■ ■":Q%=6:R%=2
3240 PRINT:G=0
3250 PRINTCHR$(B+55+(B(10)*7);
3260 PRINTCHR$(C+55+((C-1)(10)*7);":";
3270 :
3280 IFG=12THEN3240
3290 RETURN
3300 :
3310 X=A%(U%,B,C):G=G+1
3320 E$=CHR$(X/16+55+(X(160)*7)
3330 E$=E$+CHR$((XAND15)+55+((15ANDX)(10)*7):
     IFX(320RX)1220RX=34THENX=32
3340 X=X-32*((64(XANDX(96)-(96(XAND123)X))
3350 PRINT"■";E$;"■";CHR$(X);:RETURN
3360 :
3370 PRINT" * = _INE CORREZIONE"CHR$(19):G=0
3380 X=48
3390 G=GAND255:Y=1100+INT(G/12+2)*4+G*3:POKEY,127ANDPEEK(Y)
3400 FORC=0TO10:NEXT:A$=""
3410 GETA$:POKEY,128OR(PEEK(Y))
3420 IFA$=CHR$(29)THENG=G+1:GOTO3380
3430 IFA$=CHR$(64)THENGOSUB4250
3440 IFA$=CHR$(157)THENG=G-1:GOTO3380
3450 IFA$=CHR$(17)THENG=G+12:GOTO3380
3460 IFA$=CHR$(145)THENG=G-12:GOTO3380
3470 IFA$=CHR$(19)THENG=0:GOTO3380
3480 IFA$=CHR$(42)THENA=X%+2:RETURN
3490 IFA$("0"OR(A$)"9"ANDA$("A")ORA$)"F"THEN3390
3500 POKEY-1,128OR(X+64*(X)64))
3510 T=ASC(A$)+(ASC(A$))60)*7-48+(16*(X+(X)60)*7-48)):X=ASC(A$)
3520 POKEY,128OR(X+64*(X)64))
3530 A%(U%,G/16,(GAND15))=T
3540 POKEY+1,T*(T)32)*(T(96)+(T-96)*(T)96)*((T(128))-
     -32*((T(33)+(T)127))
3550 GOTO3390
3560 :
3570 REM ---- MODIFICHE AL DISCO ----
3580 CLOSE1:CLOSE2:PRINT"■■—AMBIO DEL DISCHETTO "
3590 GOSUB780
3600 OPEN1,8,15:OPEN2,8,2,"#"
3610 IFA$)"E"ORA$("A"THEN2630
3620 K%=0:L%=0:N%=2:M%=0:O%=0:P%=4:Q%=9:R%=2:T%=18:U%=0
3630 V%=0:W%=0:X%=0:Y%=15:Z%=15
3640 ONASC(A$)-65GOTO3820,3850,3880,3960
3650 REM ---- DIR SEND ----
3660 K%=4:L%=4:M%=6:N%=0:P%=5:Q%=10:U%=1:V%=0:W%=2:X%=H%
     :Y%=7:Z%=31:X=1:R%=2
3670 PRINTCHR$(147)"■■■■■ \EMORIZZARE —IRECTORY"
3680 I%=2:GOSUB4000:GOTO570
3690 :
3700 G=G%(X)-1:IFG%(0)(2THENG=X-1
3710 GOTO590
3720 :
3730 IFG=-1THENE$=E$+CHR$(0):RETURN
3740 E$=E$+CHR$(A%(G/8+1,(GAND7)*2+C/16,15ANDC)):RETURN
3750 :
3760 X=X+1:I%=I%+32:GOTO630
```

```
3770 :
3780 IFX>G%(0)THENPRINT#1,"B-P 2";0:PRINT#2,CHR$(0)+CHR$(255);
3790 GOTO710
3800 :
3810 REM ---- DIR BZ.SEND ----
3820 PRINTCHR$(147)"■■■■■■  ─ARE UN NUOVO NOME AL DISCO
3830 V%=9:S%=0:I%=144:PRINT#1,"U1 2 0 18 0":GOTO590
3840 REM ---- BAM SEND ----
3850 PRINTCHR$(147)"■■■■■■       MEMORIZZARE    |♠\
3860 Y%=8:S%=0:PRINT#1,"U1 2 0 18 0":GOTO570
3870 REM ---- DEF. BLOCK ----
3880 PRINTCHR$(147)"■■■■■■  \EMORIZZARE"T%;S%" BLOCCHI
3890 GOSUB2980:IFA$="*"ORT%=0THEN2630
3900 U%=C%(T%,S%)-1:IFU%<0THEN2640
3910 PRINT"■DESTINAZIONE":T%=0
3920 GOSUB2980:IFT%>0THEN600
3930 S%=C%(0,U%)AND63:T%=INT(C%(0,U%)/64):GOTO590
3940 :
3950 REM ---- TUTTO ----
3960 PRINTCHR$(147)"■■■■■■UTTI I BLOCCHI CARICATI VENGONO MEMORIZZATI.
3970 K%=5:U%=0:X%=E%:RETURN
3980 :
3990 GOSUB4000:GOTO590
4000 S%=C%(0,A)AND63:T%=INT(C%(0,A)/64):RETURN
4010 :
4020 R%=2:RETURN
4030 :
4040 REM ---- FILE ----
4050 J%=0:K%=2:L%=3:N%=4:M%=0:O%=0:P%=0:Q%=7:R%=2
4060 U%=0:V%=0:W%=0:X%=336:Y%=54:Z%=0
4070 PRINTCHR$(147)"■ER FAVORE INTRODUCETE IL NUMERO DEL PROGRAMMA
4080 INPUT"■     (VEDI  ─IRECTORY):";A$
4090 PRINTCHR$(147)"      ■ _♠─┐♠■ / ■♥┐||┌_┐■  _ILE /."A$
4100 PRINT"----------------------------------------
4110 G=VAL(A$)-1:IFG<10RG)=H%*8THEN2630
4120 T%=A%(1+INT(G/8),2*(GAND7),3):S%=A%(1+INT(G/8),2*(GAND7),4)
4130 GOTO4170
4140 :
4150 GET#2,E$:T%=ASC(E$+CHR$(0))
4160 GET#2,E$:S%=ASC(E$+CHR$(0))
4170 IFT%=0THEN4220
4180 IFT%<10THENPRINT" ";
4190 PRINTT%;"/";
4200 IFS%<10THENPRINT" ";
4210 PRINTS%;",";:RETURN
4220 PRINT:PRINT"■ ┐L FILE HA"254*(A*Y%+B)+S%-2" |YTE
4230 A=1+X%:B=Y%+1:RETURN
4240 :
4250 REM ----  STAMPA  ----
4260 FORD=0TO24:E$="":FORG=0TO39
4270 X=127ANDPEEK(D*40+G+1024):E$=E$+CHR$((X<32)*-96+X)
4280 NEXT:PRINT#11,E$+CHR$(14):NEXT:RETURN
4290 :
4300 REM ---- MESSAGGI DI ERRORE ---
4310 FORA=0TO9:READF$(A):NEXT
4320 IFPEEK(49152)=162THENRETURN
4330 :
4340 DATA"|LOCK-|EADER NON TROVATO"
4350 DATA"─ARATTERE ♥┐/─ NON TROVATO
```

```
4360 DATA"|LOCCO DATI NON PRESENTE
4370 DATA"¬RRORE NELLA SOMMA DI CONTROLLO
4380 DATA "DECODIFICA ERRATA DEL BYTE
4390 DATA"¬ATI NEL BUFFER <> DATI SU DISCO
4400 DATA"¬ROTEZIONE DA SCRITTURA
4410 DATA"¬RRORE NELLA SOMMA DI CONTROLLO DELL'|EADER
4420 DATA"┐L BLOCCO DATI E' TROPPO LUNGO
4430 DATA"└'┐¬ E' DANNEGGIATO
4440 :
4450 REM --- MP ----
4460 FORD=0TO66:READG:POKE12*4096+D,G:NEXT:RETURN
4470 DATA162,248,173,80,194,24,74,168
4480 DATA169,88,176,2,169,46,157,88,194
4490 DATA152,24,232,208,239,162,248,173
4500 DATA81,194,24,74,168,169,88,176,2
4510 DATA169,46,157,96,194,152,24,232
4520 DATA208,239,162,248,173,82,194,24
4530 DATA74,168,169,88,176,2,169,46,157
4540 DATA104,194,152,24,232,208,239,96
22555 Q6Q7Q8Q9Q20
```

CHIP Monitor

Lunghezza del programma: 4100 byte. Linguaggio di programmazione: Assembler. Accessori: drive o registratore.

Chi vuole sfruttare appieno le possibilità di un computer e capire il suo funzionamento, deve conoscere il codice macchina. In questo articolo presentiamo un sistema di approccio al codice macchina del Commodore 64.
Finché un programmatore sviluppa dei programmi BASIC, può ignorare completamente la vita interna del calcolatore. Per esempio non deve sapere con quale processore lavora il computer. Perché dunque dovrebbe imparare il codice macchina?
Per esempio si utilizza un programma il cui funzionamento è comprensibile solo fino alla parte scritta in codice macchina. Oppure, sviluppando un programma, si constata che, nonostante l'impiego di tutti i mezzi offerti dal linguaggio scelto, il programma lavora troppo lentamente. L'unica soluzione spesso è la traduzione delle parti più lente in codice macchina.
Un altro esempio è il caso in cui il calcolatore deve eseguire delle funzioni di controllo; il linguaggio usato però non mette a disposizione i comandi di input/output necessari, che invece sono contenuti nel codice macchina.
Se infine si desidera conoscere meglio o cambiare il funzionamento del proprio calcolatore, ciò non può avvenire senza la conoscenza del sistema operativo, che non è altro che un programma molto potente in codice macchina. È scritto in codice macchina perché il microprocessore, la mente del computer, capisce soltanto quello. Ogni microprocessore ha un proprio codice macchina.
L'importanza del codice macchina si può valutare solo se si è consapevoli del fatto che istante per istante una parte di programma viene tradotta nel codice macchina del microprocessore.
Durante l'esecuzione di un programma BASIC, è l'interprete BASIC ad avere il controllo della macchina. Per scorrere ogni linea in modo da tradurre le istruzione in BASIC prima di passarle al microprocessore perché le esegua, serve molto tempo. Perciò i programmi in codice macchina che non devono procedere a questa interpretazione delle istruzioni BASIC, sono molto più veloci.
Durante lo sviluppo di un programma, il calcolatore per la maggior parte del tempo è in un loop in codice macchina, in cui viene costantemente richiesto l'input da tastiera. Nella parte restante di tempo invece viene interpretato un comando o viene interpretata ed eseguita una linea del programma BASIC.

L'alternativa è che la macchina lavori con un programma in codice macchina registato nella memoria di lavoro dall'utente.
Ogni istruzione in codice macchina consiste in uno o più byte. Il Monitor permette l'accettazione veloce di programmi in codice macchina mediante introduzione della sequenza di tutti i byte in forma esadecimale (Hex-Dump).
Per sviluppare questo programma in codice macchina i Monitor migliori mettono a disposizione un Assembler line-by-line. Mediante questo Assembler è possibile introdurre i comandi in una forma mnemonica (mnemonics), che richiami la funzione svolta, quindi particolarmente pregnante. La trasformazione in byte avviene automaticamente. In seguito un Disassembler può trasformare di nuovo le sequenze di comandi in mnemonics, rendendoli più leggibili al programmatore.
Prima di poter usare il Monitor in modo sensato per l'introduzione di sequenze binarie, è meglio farsi dare delle spiegazioni da un esperto sulla struttura del microprocessore (un 6502 nel caso del Commodore 64) e sul suo set di istruzioni in forma di mnemonics. Il Monitor che vi presentiamo qui è scritto direttamente in linguaggio macchina.

Premesse al programma Monitor

Questo programma in codice macchina della lunghezza di quattro Kbyte è registrato nell'area di memoria da $C000 a $CFFF. Senz'altro nessuno vorrà usare il programma in quest'area. Dal momento però che il programma Monitor è in grado di copiare se stesso in un'altra area, questa sistemazione non presenta inconvenienti.
Il Monitor viene avviato con un "SYS(12*4096)". In seguito si annuncia visualizzando il contenuto dei registri, mentre il cursore lampeggia sull'ultima linea. A chi ha già lavorato con il TIM (Terminal Interface Monitor) della serie Commodore precedente, questo quadro sarà familiare.
A questo punto il Monitor è pronto ad accettare dei comandi.
Si comincia col memorizzare la forma finale del programma in codice macchina con il tipo header tre. Introducete la linea seguente e premete RETURN:

```
.S"RAT MON 64",01,
C000,D000
```

Adesso il Monitor viene memorizzato su cassetta. Quando invece il programma deve essere memorizzato su disco bisogna mettere 08 al posto di 01.
Il programma così memorizzato può venire caricato con un normale LOAD. Infine bisogna dare un NEW per sistemare il puntatore BASIC in una posizione conveniente. In caso contrario apparirà il messaggio di errore "?OUT OF MEMORY ERROR", anche se lo spazio libero in memoria è ancora grande. Il Monitor a questo punto può partire con SYS(12*4096)". Come nella programmazione in BASIC tutti gli input devono terminare con un RETURN.
Quando al caricamento del Monitor in memoria è già contenuto un programma che non si vuole perdere, bisogna dare l'avvio al Monitor immediatamente dopo il caricamento, e introdurre la seguente linea:

```
.ED 0001,0000
```

Con questo comando il puntatore alla fine del programma BASIC viene reinizializzato e il programma BASIC è salvato.
Tutti gli input numerici, se non è specificato altrimenti, avvengono in forma di numeri esadecimali a due o quattro cifre, senza essere necessariamente preceduti da "$". Con comandi che permettono l'input di numeri con base diversa, i numeri senza prefisso vengono comunque considerati rappresentazioni esadecimali. I numeri decimali devono essere preceduti da un "!" e quelli binari da un "%". Questa differenziazione dalla sintassi normale dell'Assembler 6502 è necessaria, per permettere la compatibilità del Commodore col TIM.
Alla prima chiamata del Monitor il vettore BRK del Commodore 64 viene diretto alla Break-Entry del Monitor, in modo che ogni comando BRK nel programma si ramifichi verso il Monitor. È possibile costringere il programma in fase di sviluppo a fermarsi in posizioni tatticamente favorevoli e controllare che tutto sia corretto in base allo status dei registri.
Quando all'avvio del Monitor il puntatore allo stack è inferiore a $50, viene portato a $F8, l'indirizzo della Call Entry è spostato nello stack e compare il messaggio "?LOW STACK ERROR".
Dopo una situazione di questo tipo una routine che finisce con RTS assume il controllo, il comando RTS trova l'indirizzo della Call Entry allo stack e torna indietro al Monitor. In questo modo si impedisce che il programma gestisca indirizzi assurdi.

Le istruzioni del Monitor

L'Assembler diretto
Sintassi del comando:
.A aaaa xxx yyyyy
Il comando xxx viene scritto all'indirizzo corrente aaaa con il suo operando yyyyy. Fondamentalmente viene impiegata la sintassi normale dell'Assembler 6502. Dal momento che non si tratta di un Assembler simbolico, tutti gli indirizzi devono essere introdotti come numeri esadecimali di due o quattro cifre. Grazie all'univocità delle basi usate, non è indispensabile, ma possibile, usare "$" per gli operandi esadecimali. Nel modo di indirizzamento immediato, oltre i valori esadecimali con o senza "$", sono permessi anche numeri binari a otto cifre riconoscibili dal prefisso del carattere ASCII "(CHR$(39)) e dal "%" che li precede. I comandi indirizzati all'accumulatore si riconoscono grazie alla "A" che li segue. L'Assembler però è in grado di riconoscere questo indirizzamento anche senza "A", dal momento che per questi comandi non esiste il modo di indirizzamento "Implied".
Per salti relativi si può indicare sia l'indirizzo di arrivo a quattro cifre che il displacement. Dopo un indirizzamento riuscito, nella linea successiva compare l'indirizzo del prossimo comando. Il cursore si trova vicino all'indirizzo, a destra. Se a questo punto viene dato un RETURN, senza introdurre un nuovo comando, l'Op Code che si trova a quell'indirizzo viene stampato disassemblato. Il cursore poi si posiziona sul primo carattere del comando disassemblato. In questo modo è possibile per esempio controllare, ed eventualmente correggere, la parte di programma introdotta.
Tutti gli ingressi sbagliati vengono riconosciuti e segnalati con un "?". L'unica eccezione è il comando STX con l'indirizzamento "Assoluto,Y". Questo comando viene tradotto dall'Asembler in modo logicamente corretto, ma non può venire eseguito dal processore.
Il Disassembler
Sintassi del comando:
.D aaaa,bbbb
L'area di memoria da aaaa a bbbb viene stampata disassemblata con il codice hex corrispondente. Se l'indirizzo finale bbbb non viene specificato, gli viene assegnato il valore $FFFF. Se l'output non avviene su una periferica mediante CMD, compaiono soltanto cinque linee. Il Disassembler poi aspetta che venga battuto un tasto.
Quando i dati devono venire inviati ad un'altra unità, per esempio una stampante, il disassemblamento avviene in modo continuato. In questo modo si facilita l'output, per esempio su stampante.
Il processo di disassemblamento può essere interrotto battendo la combinazione di tasti RUN/STOP. Se il Disassembler trova un Op Code sconosciuto, fa apparire un "???". Come nell'Assembler il comando STX con l'indirizzamento "Assoluto,Y", non viene considerato errato.
La parte esadecimale dell'output può subire delle alterazioni a piacere. Con RETURN si introducono in memoria i dati alterati. La linea alterata e quella seguente vengono nuovamente disassemblate, il cursore si trova sul primo carattere del codice Hex nella linea successiva. Per i programmatori esperti questo sistema può servire anche ad introdurre un programma.
A questo punto vengono letti tanti byte quanti ne indica il comando introdotto. I codici esadecimali devono essere separati dall'indirizzo da uno o al massimo due blank.
Displacement per salti relativi
Sintassi del comando:
.B aaaa,bbbb
Questo comando calcola il displacement per il salto relativo dall'indirizzo aaaa all'indirizzo di arrivo bbbb e fa apparire l'indirizzo finale separato da ":". Se il salto relativo da aaaa a bbbb non è possibile, compare il messaggio "?BRANCH ERROR IN aaaa".
Riempimento di un'area di memoria mediante un certo valore
Sintassi del comando:
.F aaaa,bbbb,cc
L'area di memoria da aaaa fino a bbbb incluso viene riempita con il valore cc.
Introduzione di byte o sequenze di caratteri
Sintassi del comando:
.W aaaa "text1" bb cc "text2"...
La sequenza di byte corrispondente agli output "text1", bb, cc, "text2" viene registrata all'indirizzo aaaa. I testi ASCII devono essere racchiusi tra " (CHR$(34)), si possono mescolare testi e sequenze di byte. Dopo l'esecuzione del comando nella linea successiva compare l'indirizzo finale più uno.
Generazione e alterazione di dump esadecimali
Sintassi del comando:
.M aaaa,bbbb

L'area di memoria da aaaa a bbbb viene stampata in forma esadecimale. Sullo schermo, dopo l'indirizzo del primo byte della linea, compaiono otto byte. Se i dati vengono inviati ad un'altra unità, in una linea vengono stampati sedici byte. Nell'output su stampante in questo modo si risparmia carta. Se manca l'indirizzo finale bbbb, compare una sola linea. L'elenco può venire interrotto mediante la combinazione di tasti RUN/STOP.
Quando si deve alterare un byte, si porta il cursore alla posizione corrispondente, si introduce il nuovo byte e si dà un RETURN. Quando l'alterazione non è possibile, per esempio se il byte in questione si trova nella ROM, alla posizione richiesta compare un "?". Gli input che non rappresentano un byte esadecimale (per esempio "XX"), vengono ignorati e il byte corrispondente in memoria resta inalterato.

Generazione di linee DATA

Sintassi del comando:
.ED aaaa,bbbb
I programmi in linguaggio macchina più piccoli vengono spesso memorizzati in linee DATA e scritti a richiesta nell'area di memoria corretta mediante READ e POKE. Questo comando genera una linea DATA di questo tipo dall'area di memoria aaaa al bbbb incluso. La prima linea DATA contiene il numero dell'ultima linea BASIC più uno e le linee successive assumono numeri crescenti.
Nel caso queste linee assumano numeri dubbi, apparirà il messaggio "?ILLEGAL LINE NUMBER ERROR IN xxxx". Quando per una nuova linea DATA sono a disposizione meno di 256 byte, compare il messaggio "?OUT OF MEMORY ERROR IN xxxx". In entrambi i casi l'esecuzione del comando viene interrotta, xxxx dà l'indirizzo del primo byte che non può essere utilizzato.
Durante l'esecuzione di questo comando gli indirizzi di link del programma BASIC vengono ricalcolati, il puntatore si dirige alla fine del programma e tutte le variabili BASIC vengono cancellate.

Cambiamento di base di un numero

Sintassi del comando:
.C aaaa
Il numero aaaa viene dato in sistema esadecimale, binario o decimale. La base in cui aaaa è calcolato viene indicata da un prefisso: "!" per il sistema decimale, "$" per quello esadecimale e "%" per il sistema binario. Il numero di cifre è a piacere, comunque vengono elaborati soltanto i 16 bit di valore inferiore. I numeri senza prefisso vengono interpretati come numeri esadecimali.

Semplici operazioni di calcolo a 16 bit

Sintassi del comando:
.O aaaa,bbbb,c
Per l'input dei numeri aaaa e bbbb valgono le regole specificate per il cambiamento delle basi. Separata da una virgola, segue l'operazione da eseguire. Il risultato viene sempre dato come numero esadecimale di quattro cifre.
Le operazioni possibili sono:
c = "+" aaaa e bbbb vengono sommati
c = "—" bbbb viene sottratto da aaaa
c = "*" aaaa viene moltiplicato per bbbb
c = "/" aaaa viene diviso per bbbb
c = "A" operazione di tipo AND
c = "E" operazione di tipo OR esclusivo
c = "O" operazione di tipo OR inclusivo

Ricerca di indirizzi

Sintassi del comando:
.H aaaa,bbbb,cccc,dddd
Questo comando cerca nell'area di memoria da aaaa a bbbb i comandi che indirizzano l'area da cccc e dddd. Se dddd non viene specificato, vengono cercati soltanto i comandi che riguardano l'indirizzo cccc. I comandi che soddisfano le condizioni specificate vengono disassemblati. Degli Op Code dubbi provocano l'interruzione del processo di ricerca e appare il messaggio "?ILLEGAL CODE ERROR IN xxxx".

Ricerca di sequenze di byte o caratteri

Sintassi del comando:
.EH aaaa,bbbb "text1" cc dd ecc.
Nelle aree di memoria da aaaa a bbbb viene cercata la sequenza indicata. I testi ASCII devono essere racchiusi tra " (CHR$(34)). Testi e byte si possono mescolare. Quando la sequenza desiderata viene trovata, compare il suo indirizzo iniziale.

Spostamento di blocchi di memoria

Sintassi del comando:
.T aaaa,bbbb,cccc—
oppure:

.T aaaa,bbbb,cccc+
L'area di memoria da aaaa e bbbb incluso viene spostata nell'area con indirizzo finale o iniziale cccc. Quando cccc è l'indirizzo iniziale è seguita da "—"; quando invece è l'indirizzo finale viene seguita da "+".

Adattamento del codice macchina ad un'altra area

Sintassi del comando:
.ER aaaa,bbbb,cccc,dddd,eeee—
oppure:
.ER aaaa,bbbb,cccc,dddd,eeee+
Questo comando dice al programma che occupa l'area di memoria da aaaa a bbbb, che l'area di memoria da cccc a dddd deve essere spostata all'area con indirizzo iniziale o finale eeee. Se eeee è l'indirizzo iniziale, segue un "—"; se invece è l'indirizzo finale, segue un "+".
Nella correzione degli indirizzi si tiene conto di tutti i possibili sistemi di indirizzamento. Vengono riconosciuti e segnalati insieme al relativo indirizzo i seguenti errori.
Si cerca di assegnare ad un comando di due byte un indirizzo $FF più grande. In questo caso l'indirizzo non viene cambiato e appare il messaggio "?ZERO PAGE ERROR IN xxxx". L'adattamento dell'indirizzo tuttavia non viene interrotto.
Quando, in conseguenza dello spostamento, un salto relativo avverrà ad una distanza troppo grande, compare il messaggio "?BRANCH ERROR IN xxxx" e il byte non viene cambiato. L'esecuzione del comando non viene interrotta.
Se .ER trova un Op Code dubbio, appare il messaggio "?ILLEGAL CODE ERROR IN xxxx" e l'esecuzione del comando viene subito interrotta. In questo modo si evita che il comando penetri nelle tabelle dati e le danneggi.

Aggiunta di byte nel codice macchina

Sintassi del comando:
.I aaaa,bbbb,cccc,dd
Nel programma nell'area da aaaa a bbbb prima del comando cccc viene fatto posto per dd byte. Perciò l'area cccc fino a bbbb viene spostata di dd byte verso l'alto, e lo spazio ottenuto viene occupato da $EA (NOP). Si possono aggiungere 255 byte al massimo.
Quando alla fine del programma c'è una tabella dati coinvolta nello spostamento, il codice $FF deve venire inserito immediatamente prima di questa, e per bbbb bisogna stabilire l'indirizzo finale. Il codice $FF provoca un'interruzione dell'adattamento degli indirizzi prima della tabella.
Gli indirizzi di eventuali errori comunque si riferiscono alla vecchia disposizione. Nel caso di errori con indirizzo superiore a cccc bisogna aggiungere il valore dd.

Cancellazione di byte dal programma in codice macchina

Sintassi del comando:
.K aaaa,bbbb,cccc,dd
Dal programma che occupa l'area da aaaa a bbbb, all'indirizzo cccc vengono soppressi dd byte. Perciò l'area da cccc+dd a bbbb viene spostata in basso di dd byte. Possono venire cancellati 255 byte al massimo, per il resto valgono gli stessi principi spiegati per il comando .I.

Avvio di un programma in codice macchina

Sintassi del comando:
.G aaaa
Un programma in codice macchina parte dall'indirizzo aaaa. Quando aaaa non viene specificato, si usa come indirizzo iniziale lo status del contatore di programma memorizzato temporaneamente. Un programma che è stato interrotto mediante BRK, può venire continuato con .G, senza specificare l'indirizzo aaaa. All'avvio di un programma mediante questo comando, i contenuti dei registri temporaneamente memorizzati vengono trasmessi al programma.

Controllo e alterazione del contenuto di un registro

Sintassi del comando:
.R
I contenuti temporaneamente memorizzati del registro vengono visualizzati:

PC	IRQ	NV-BDIZC	AC	XR	YR	SP
E147	EA31	00110000	00	00	00	F8

Le abbreviazioni usate hanno i seguenti significati:

PC	contatore di programma
IRQ	vettore IRQ
NV-BDIZC	registro di status del processore
AC	accumulatore
XR	registro X
YR	registro Y

SP status del puntatore allo stack
I contenuti del registro possono venire alterati sullo schermo e introdotti mediante RETURN nella memoria temporanea. Questi contenuti diventeranno effettivi non appena il programma sarà avviato mediante .G.

Memorizzazione di programmi in codice macchina

Sintassi del comando:
.S "name",aa,bbbb,cccc
L'area di memoria da bbbb a cccc incluso viene memorizzata col nome "name" sull'unità aa. Quando aa è uguale a $08, la memorizzazione avviene su disco, quando invece aa corrisponde a $01, la memorizzazione avviene su cassetta. Bisogna fare attenzione che cccc corrisponda all'indirizzo finale più uno dell'area da memorizzare.
Il nome indicato può avere una lunghezza da 0 a 15 caratteri e deve essere racchiuso tra " (CHR$(34)), tutte le specifiche devono essere separate da virgole. I programmi memorizzati in questo modo ricevono il tipo Header tre, cioè vengono caricati sempre all'indirizzo originario. I calcolatori della serie 2/3/4 e 8000 non ammettono questo tipo e perciò non sono in grado di caricare i programmi memorizzati in questo modo.

Caricamento di programmi in codice macchina

Sintassi del comando:
.L "name",aa
Questo comando carica il programma memorizzato col nome "name" dall'unità aa. Indipendentemente dal tipo Header, il programma viene caricato sempre in modo assoluto. Non viene utilizzato il puntatore BASIC. La specificazione di aa si può omettere nel caso di programmi su cassetta. Se invece manca la specificazione del nome del programma, viene caricato il primo programma sulla cassetta.

Ritorno al BASIC

Sintassi del comando:
.X
Si verifica un ritorno all'interprete BASIC
(READY.), i contenuti dei registri temporaneamente memorizzati non vengono stampati.

Istruzioni per l'uso

Quando si vogliono far funzionare contemporaneamente programmi in BASIC e programmi in codice macchina caricati nell'area BASIC, bisogna riservare e proteggere lo spazio per il programma in codice macchina. Se il programma in codice macchina si trova all'estremità superiore della BASIC-RAM, bisogna spostare nel modo desiderato il puntatore alla fine della BASIC-RAM. Nelle celle di memoria $37 e $38 è memorizzato il limite superiore della RAM a disposizione del BASIC. A questo punto non basta cambiare solo queste celle, ma bisogna anche assicurarsi che il puntatore alla fine dell'ultima stringa memorizzata, che si trova nelle celle $33 e $34, non indichi il settore protetto. Il modo più semplice per evitarlo è il comando CLR.
Ecco un esempio:
la linea che segue protegge dal BASIC la memoria dall'indirizzo $9000 in poi:
0 POKE45,0:POKE46,144:CLR
L'area riservata al BASIC nella memoria di lavoro si può comunque restringere anche mediante il Monitor:
:M 0033 .:033 00 A0 00 00 00 A0 00 FF
Per proteggere la memoria dall'inidirizzo $9000 dovete cambiare i byte sottolineati nel modo seguente:
.:0033 00 90 00 00 00 90 00 FF
Per essere preparati ad eventuali nuove versioni della BASIC-ROM sono stati usati il meno possibile gli indirizzi specifici di sistema.
Della ROM sono state usate soltanto delle routine i cui indirizzi molto verosimilmente non cambieranno neanche nelle versioni successive. Non si è potuto comunque rinunciare ad alcuni indirizzi RAM (Tab. 2), ma anche in questo caso si tratta di una disposizione che probabilmente non cambierà. Inoltre è stato riportato un elenco degli indirizzi dei salti e delle tabelle del Monitor (Tab. 1) e un elenco della sua memoria di lavoro (Tab. 3).

Ulrich Roller

C000 Entry del Monitor
C029 Break Entry del Monitor
C07F salto nel loop di richiesta di comandi del Monitor. Viene creata una nuova linea sul video e il puntatore viene resettato
C08E salto diretto alla richiesta di comandi, il puntatore non viene resettato e non viene creata una nuova linea
C09E riconoscimento del comando
C0C3 tabella degli indirizzi dei comandi
C10E legge un carattere dallo schermo; quando si arriva alla fine dell'input, viene cancellato il Carry e trasmesso un $20
C11D legge un byte esadecimale dallo schermo e lo trasmette all'accumulatore. Viene saltato al massimo uno spazio. Se non si trova nessun byte esadecimale, viene cancellato il Carry
C124 come C11D, però subito il primo carattere deve essere una cifra esadecimale
C144 legge un indirizzo con un numero a piacere. L'indirizzo letto viene registrato in P1; se non si trova un indirizzo valido, il Carry viene cancellato e il valore di P1 non cambia
C154 tralascia due caratteri a piacere, il Carry viene cancellato
C162 trasforma una cifra esadecimale in un numero
C16D legge due indirizzi espressi in una numerazione a piacere. Se il secondo indirizzo è mancante, P2=P1 e il Carry viene cancellato
C17D scambia P1 con P2
C18D salto alla routine di errore "?"
C192 indica in esadecimale il valore di P1
C19D indica in esadecimale il carattere dell'accumulatore
C1AC WRTWO: indica il carattere nel registro X e nell'accumulatore
C1B5 trasforma il valore dell'accumulatore in un numero esadecimale
C1BF porta gli indirizzi di P1 a $0200/0201
C1CA stampa tanti byte in esadecimale, quanti ne indica il valore dell'accumulatore. Il puntatore del byte da stampare è P1. P1 viene incrementato
C1DE stampa due blank
C1E4 esegue un ritorno di carrello
C1E9 incrementa P1
C1F4 crea una nuova linea sul video con il numero in Y
C202 legge un byte esadecimale dallo schermo, lo scrive all'indirizzo indicato da P1, controlla se il byte è stato registrato esattamente e in caso contrario segnala un errore. Se sullo schermo non viene trovato alcun byte valido, il contenuto della cella di memoria indicata da P1 non viene cambiato. Infine il valore di P1 viene innalzato di 1 e il valore di $B5 viene diminuito di 1
C218 prepara l'input/output del contenuto dei registri
C223 dà il valore dell'accumulatore in forma di numero binario a otto cifre preceduto da un blank
C238 determina le lunghezze dei comandi e il modo di indirizzamento dell'Op Code nell'accumulatore. Il registro X contiene la lunghezza dei comandi e il registro A il modo di indirizzamento. Quando l'Op Code è dubbio,è settato il Carry, X=1 e A=$0F.
Modi di indirizzamento:

00 = implicito	05 = (indiretto,X)	0A = assoluto,X
01 = accumulatore	06 = (indiretto,Y)	0B = assoluto,Y
02 = immediato	07 = pagina zero	0C = relativo
03 = pagina zero,X	08 = indiretto	0F = illegale
04 = pagina zero,Y	09 = assoluto	

C25C dà il comando indicato da P1 disassemblato col relativo codice esadecimale
C288 dà il comando indicato da P1 disassemblato, non viene creata una nuova linea
C304 calcola l'indirizzo di arrivo del salto relativo indicato da P1 e lo indica in esadecimale
C367 calcola l'indirizzo di arrivo del salto relativo indicato da P1 e lo indica in Y, X (L, H)
C381 determina la lunghezza e l'indirizzo del comando indicato da P1. Per comandi senza indirizzo, il Carry viene cancellato. Per Op Code considerati dubbi, appare il messaggio di errore "?ILLEGAL CODE ERROR IN xxxx" e si salta alla routine preposta alla routine da chiamare. A contiene la lunghezza del comando, X e Y l'indirizzo
C3A2 crea una nuova linea e dà un messaggio di errore con indice X
C3AD dà un ERROR
C3AF dà il testo in CE28 puntato dal registro X
00: (CR)----PC IRQ NV-BDIZC AC XR SP
23:—ERROR
29:—IN—
2D:MISSING+/—
38:LOW STACK
41:ILLEGAL CODE
4D:FOUND

55:BRANCH
5B:ZERO PAGE
64:OUT OF MEMORY
71:ILLEGAL LINE NUMBER
84:LOAD

C3BD dà il messaggio "?ILLEGAL CODE ERROR IN xxxx"
C3BF messaggio di errore puntato dal registro X con indirizzo
C3CF messaggio di errore puntato dal registro X senza indirizzo
C3D4 legge un indirizzo esadecimale dallo schermo e lo registra in P3
C3E5 trattamento dell'errore "?"
C3EC legge l'indirizzo iniziale e finale di un blocco sorgente e li registra in P1 e P2. Vengono letti inoltre l'indirizzo iniziale e quello finale del blocco di arrivo. Se il blocco di arrivo è più in alto del blocco sorgente, P3 contiene l'indirizzo finale del blocco di arrivo e è settato il Carry. In caso contrario il Carry viene azzerato e P3 contiene l'indirizzo iniziale del blocco di arrivo
C444 incrementa P1 della lunghezza del comando che P1 indica
C452 controlla se P1 è maggiore di P2. In caso affermativo si salta al loop di input, altrimenti si torna indietro alla routine da chiamare
C466 routine di spostamento per la routine C3EC
Quando il Carry è azzerato, P3 contiene l'indirizzo iniziale del blocco di arrivo e viene spostato verso l'alto
C498 legge un numero preceduto da un carattere a piacere e da un prefisso per il riconoscimento della base. Se il prefisso manca, viene letto un numero esadecimale. Il prefisso di riconoscimento viene memorizzato in $B4
C527 converte il valore di P1 in decimale, X contiene la lunghezza meno uno del decimale ottenuto, le cifre di questo numero vengono registrate da $0100 in ordine inverso
C573 controllo ed esecuzione di STOP
C57C cancella una linea di schermo, disassembla il comando indicato da P1 e posiziona il cursore dietro l'indirizzo dato
C590 scambia P1 con P3 e P2 con P4
C59E controlla se l'indirizzo in X,Y si trova nell'area limitata da P3 e P4. In caso contrario viene settato il Carry
C5B2 controlla se P1 si trova nell'area limitata da P3 e P4. In caso contrario viene settato il Carry
C5C2 dà il byte di arrivo del comando di salto indicato da P1 all'indirizzo memorizzato in X, Y. Se la distanza è troppo grande, compare il messaggio "?BRANCH ERROR IN xxxx", viene settato il Carry e si torna indietro alla routine da chiamare. A contiene il byte di arrivo, il registro Y contiene $01
C5DD fa apparire un messaggio di errore con indice X con il relativo indirizzo e setta il Carry
C5EA aumenta X e Y secondo il valore di P5. Quando il byte di valore più elevato del nuovo indirizzo è $00, viene sistemato un flag zero
C5F7 fa apparire il valore di P1 in decimale
C605 legge un numero binario a 8 bit. Se durante la lettura si incontra un errore, il Carry è azzerato
C61D legge un carattere sullo schermo e controlla se si tratta di un valore esadecimale, in caso affermativo setta il Carry
C632 completo adattamento per .I, .K. P1 e P2 indicano l'area in cui devono essere sistemati gli indirizzi. P3 e P4 limitano l'area da spostare e P5 indica di quanto si deve spostare. Se si deve spostare verso il basso, P5 deve contenere il complemento a due della differenza degli indirizzi
C6C1 legge una sequenza di byte o di caratteri dallo schermo e la registra all'indirizzo indicato da P1
C6F5 comando ".W"
C709 comando ".I" oppure "K"
C79C comando ".ER"
C7DA comando ".ED"
C8E4 comando ".R"
C918 comando ".M"
C936 comando ".;" (routine di alterazione per ".R")
C95D comando ".:" (routine di alterazione per ".M")
C977 comando ".D"
C9A3 comando "." (routine di alterazione per ".D")
C9D8 comando ".F"
C9F2 comando ".T"
C9F8 comando ".C"
CA2F comando ".G"
CA5E comando ".X"
CA67 comando ".H"
CA98 comando ".EH"
CADA comando ".B"
CAF6 comando ".O"

CBC9	comando ".L" oppure ".S"
CC7A	comando ".A"

Tab. 1: i più importanti indirizzi di salto e di tabelle del Monitor

0013	unità di I/O attiva
0016	puntatore allo stack di stringa
002B-002C	puntatore all'inizio del programma BASIC
002D-002E	puntatore all'inizio delle variabili
002F-0030	puntatore all'inizio degli array
0031-0032	puntatore alla fine degli array
0033-0034	puntatore all'inizio dell'ultima stringa usata
0037-0038	puntatore alla fine della BASIC-RAM
009D	flag per il modo diretto
00C6	numero dei tasti battuti (buffer della tastiera)
00D0	flag-CR
00D1-00D2	puntatore all'inizio della linea sullo schermo
00D3	colonna del cursore
0100-0105	buffer per il cambiamento delle basi dei numeri
0200-0250	buffer di input del BASIC
0277-0280	buffer della tastiera
0300-0301	vettore per i messaggi di errore
030C-030F	registro per il comando SYS
0314-0315	vettore IRQ
0316-0317	vettore BRK
FFB7	prende la parola di status dell'accumulatore
FFBA	stabilisce i parametri del file
FFBD	stabilisce i parametri del nome di file
FFCF	BASIN: input del carattere nell'accumulatore
FFD2	BSOUT: fa apparire il carattere contenuto nell'accumulatore
FFD5	LOAD: carica un programma in memoria
FFD8	SAVE: memorizza un programma
FFE1	test al tasto STOP
FFE4	GET: prende un carattere dalla tastiera

Tab. 2: elenco degli indirizi RAM usati

00B4	memoria temporanea per scopi diversi
00B5	memoria temporanea per scopi diversi
0057 0058	puntatore 5 (P5)
005C 005D	puntatore 3 (P3)
005E 005F	puntatore 4 (P4)
00FB 00FC	puntatore 1 (P1)
00FD 00FE	puntatore 2 (P2)
0100 0105	buffer per il cambiamento di base
0200 0201	memoria temporanea per il contatore di programma
0202	memoria temporanea, status del processore
0203	memoria temporanea, accumulatore
0204	memoria temporanea, registro X
0205	memoria temporanea, registro Y

0206	memoria temporanea, puntatore allo stack
0207 0208	memoria temporanea, vettore IRQ
0209	situazione dello SP dopo un messaggio di errore del Monitor
020A 0250	buffer per scopi diversi

Tab. 3: la memoria di lavoro del Monitor

```
CHIP-MON $C000
PER CHIP SPECIAL
DI ULRICH ROLLER

INDIRIZZO INIZIALE : 49152
INDIRIZZO FINALE   : 53248

C000:EA,08,48,8A,48,98,48,BA,03A6
C008:FE,05,01,D0,03,FE,06,01,02DC
C010:AD,0C,C1,8D,16,03,AD,0D,02DA
C018:C1,8D,17,03,EA,EA,EA,EA,0510
C020:EA,EA,EA,EA,EA,EA,A9,43,0668
C028:2C,A9,42,85,B5,4A,EA,EA,046F
C030:D8,A2,03,68,9D,02,02,CA,0350
C038:10,F9,68,69,FF,8D,01,02,0369
C040:68,69,FF,8D,00,02,AD,14,0320
C048:03,8D,0E,02,AD,15,03,8D,01EC
C050:07,02,BA,8E,06,02,8E,09,01F0
C058:02,58,E0,50,B0,13,A2,38,0327
C060:20,A2,C3,A2,FA,9A,AD,0A,0472
C068:C1,48,AD,09,C1,48,4C,52,0366
C070:C0,20,E4,C1,A6,B5,A9,2A,04B3
C078:20,AC,C1,A9,52,D0,12,A9,0413
C080:00,85,A4,AE,09,02,9A,A9,0325
C088:2E,A2,0D,20,AC,C1,20,CF,0359
C090:FF,C9,0D,F0,F2,C9,2E,F0,059E
C098:F5,C9,20,F0,F1,A2,13,C9,053D
C0A0:45,D0,05,20,CF,FF,A2,16,03C0
C0A8:DD,B0,CE,F0,06,CA,10,F8,0523
C0B0:4C,E5,C3,86,B4,8A,0A,65,0427
C0B8:B4,AA,BD,C5,C0,48,BD,C4,0569
C0C0:C0,48,60,2C,E3,C8,2C,17,0382
C0C8:C9,2C,35,C9,2C,5C,C9,2C,0370
C0D0:76,C9,2C,A2,C9,2C,D7,C9,04A2
C0D8:2C,F1,C9,2C,F7,C9,2C,66,0464
C0E0:CA,2C,2E,CA,2C,5D,CA,2C,036D
C0E8:D9,CA,2C,C8,CB,2C,C8,CB,0521
C0F0:2C,F5,CA,2C,72,CC,2C,08,0389
C0F8:C7,2C,08,C7,2C,F4,C6,2C,03D4
C100:9B,C7,2C,97,CA,2C,D9,C7,04BB
C108:2C,00,C0,2C,29,C0,A5,D0,0376
C110:F0,07,20,CF,FF,C9,0D,D0,048B
C118:29,A9,20,18,60,20,1D,C6,026D
C120:EA,EA,B0,07,20,1D,C6,EA,0478
C128:EA,90,32,20,62,C1,0A,0A,0303
C130:0A,0A,8D,00,01,20,1D,C6,01A5
C138:EA,EA,90,DF,20,62,C1,0D,0493
C140:00,01,38,60,20,0E,C1,20,01A8
C148:1D,C1,90,0D,48,20,24,C1,02C8
C150:90,06,85,FB,68,85,FC,60,045F
C158:68,60,20,0E,C1,20,0E,C1,02A6
C160:18,60,C9,3A,08,29,0F,28,01E3
C168:90,02,69,08,60,20,44,C1,0288
C170:90,1B,A5,FB,85,FD,A5,FC,056E
C178:85,FE,20,44,C1,A2,02,B5,0401
C180:FA,48,B5,FC,95,FA,68,95,057F
C188:FC,CA,D0,F3,60,A9,3F,4C,051D
C190:E5,C3,A2,01,B5,FA,48,B5,04F7
C198:FB,20,9D,C1,68,48,4A,4A,03BD
C1A0:4A,4A,20,B5,C1,AA,68,29,0365
C1A8:0F,20,B5,C1,48,8A,20,D2,0369
C1B0:FF,68,4C,D2,FF,18,69,F6,04FB
C1B8:90,02,69,06,69,3A,60,A5,02A9
C1C0:FB,8D,01,02,A5,FC,8D,00,03B9
C1C8:02,60,85,B5,A0,00,20,E1,033D
C1D0:C1,B1,FB,20,9D,C1,20,E9,04F4
C1D8:C1,C6,B5,D0,F1,60,20,E1,055E
C1E0:C1,A9,20,2C,A9,0D,4C,D2,038A
C1E8:FF,E6,FB,D0,06,E6,FC,D0,0668
C1F0:02,E6,A4,60,98,48,20,E4,03D0
C1F8:C1,68,A2,2E,20,AC,C1,4C,03D2
C200:DE,C1,20,24,C1,90,0B,A2,03E1
C208:00,81,FB,C1,FB,F0,03,4C,0477
C210:8D,C1,20,E9,C1,C6,B5,60,04F3
C218:A9,03,85,FB,A9,02,85,FC,0458
C220:A9,04,60,48,20,E1,C1,68,037F
C228:A2,07,85,B5,A9,18,06,B5,035F
C230:2A,20,D2,FF,CA,10,F5,60,044A
C238:85,B5,A2,00,A5,B5,3D,D4,0447
C240:CF,5D,C0,CF,F0,07,E8,E0,057A
C248:14,90,F1,A2,0E,BD,E8,CF,04B9
C250:48,4A,4A,4A,4A,AA,68,29,02AB
C258:0F,C9,0E,60,A0,2C,20,F4,0326
C260:C1,20,92,C1,20,E1,C1,A0,0496
C268:00,B1,FB,20,38,C2,86,B5,0401
C270:20,E1,C1,B1,FB,20,9D,C1,04EC
C278:C8,C6,B5,D0,F3,20,E1,C1,05C8
C280:20,DE,C1,C8,C0,04,90,F5,04D0
C288:A0,00,B1,FB,A2,1F,DD,C8,04B2
C290:CE,F0,1A,CA,10,F8,A2,17,0463
C298:B1,FB,3D,00,CF,5D,E8,CE,04CB
C2A0:F0,06,CA,10,F3,4C,5C,C3,042E
C2A8:8A,18,69,20,AA,8A,85,B5,0399
C2B0:0A,65,B5,AA,A0,03,BD,18,0346
C2B8:CF,20,D2,FF,E8,88,D0,F6,05F6
C2C0:B1,FB,20,38,C2,B0,DE,A8,04FC
C2C8:A9,20,20,D2,FF,C0,08,F0,0472
C2D0:25,C0,09,B0,26,A9,24,C0,0351
```

```
C2D8:Ø2,DØ,Ø5,A9,23,2Ø,D2,FF,Ø394
C2EØ:A9,24,CØ,ØØ,DØ,Ø2,A9,2Ø,Ø328
C2E8:CØ,Ø1,DØ,Ø2,A9,41,CØ,Ø5,Ø342
C2FØ:FØ,Ø4,CØ,Ø6,DØ,Ø7,A9,28,Ø362
C2F8:2Ø,D2,FF,A9,24,2Ø,D2,FF,Ø4AF
C3ØØ:CØ,ØC,DØ,ØB,2Ø,67,C3,8A,Ø37B
C3Ø8:2Ø,9D,C1,98,4C,9D,C1,CA,Ø48A
C31Ø:FØ,11,CA,FØ,Ø7,AØ,Ø2,B1,Ø415
C318:FB,2Ø,9D,C1,AØ,Ø1,B1,FB,Ø4C6
C32Ø:2Ø,9D,C1,AØ,ØØ,B1,FB,2Ø,Ø3EA
C328:38,C2,C9,Ø3,FØ,27,C9,Ø4,Ø3AA
C33Ø:FØ,2Ø,C9,Ø5,FØ,11,C9,Ø6,Ø3AE
C338:FØ,15,C9,ØA,FØ,17,C9,ØB,Ø3B3
C34Ø:FØ,1Ø,C9,Ø8,FØ,Ø4,6Ø,2Ø,Ø345
C348:55,C3,A9,29,4C,D2,FF,2Ø,Ø427
C35Ø:4A,C3,A9,59,2C,A9,58,A2,Ø3DE
C358:2C,4C,AC,C1,A2,Ø2,A9,3F,Ø371
C36Ø:2Ø,D2,FF,CA,1Ø,FA,6Ø,A6,Ø4CB
C368:FC,AØ,Ø1,B1,FB,1Ø,Ø1,CA,Ø424
C37Ø:18,69,Ø2,9Ø,Ø1,E8,18,65,Ø279
C378:FB,9Ø,Ø1,E8,A8,A9,82,38,Ø47F
C38Ø:6Ø,AØ,ØØ,B1,FB,2Ø,38,C2,Ø3C6
C388:BØ,33,C9,ØC,FØ,D9,C9,Ø3,Ø44D
C39Ø:8A,48,C8,B1,FB,48,CA,CA,Ø522
C398:FØ,Ø4,C8,B1,FB,AA,68,A8,Ø522
C3AØ:68,6Ø,2Ø,E4,C1,A9,3F,2Ø,Ø395
C3A8:D2,FF,2Ø,AF,C3,A2,23,BD,Ø4E5
C3BØ:28,CE,Ø8,29,7F,2Ø,D2,FF,Ø397
C3B8:E8,28,1Ø,F3,6Ø,A2,41,2Ø,Ø376
C3CØ:A2,C3,A2,29,2Ø,AF,C3,2Ø,Ø3E2
C3C8:92,C1,4C,7F,CØ,A2,2D,2Ø,Ø3CD
C3DØ:A2,C3,3Ø,F6,2Ø,ØE,C1,2Ø,Ø39A
C3D8:1D,C1,9Ø,Ø9,85,5D,2Ø,24,Ø29D
C3EØ:C1,85,5C,BØ,5E,A9,3F,2Ø,Ø3B8
C3E8:D2,FF,DØ,DE,A9,2Ø,A4,D3,Ø5BF
C3FØ:88,91,D1,2Ø,6D,C1,9Ø,ED,Ø4B5
C3F8:2Ø,D4,C3,2Ø,ØE,C1,C9,2D,Ø39C
C4ØØ:FØ,21,C9,2B,DØ,C7,A5,5C,Ø49D
C4Ø8:AA,A5,5D,A8,A5,FD,E5,FB,Ø5D6
C41Ø:85,5C,A5,FE,E5,FC,85,5D,Ø547
C418:8A,38,E5,5C,85,5C,98,E5,Ø461
C42Ø:5D,85,5D,A5,5C,C5,FB,A5,Ø4A5
C428:5D,E5,FC,9Ø,16,A5,FD,E5,Ø56B
C43Ø:FB,AA,A5,FE,E5,FC,A8,8A,Ø65B
C438:18,65,5C,85,5C,98,65,5D,Ø314
C44Ø:85,5D,38,6Ø,AØ,ØØ,B1,FB,Ø3C6
C448:2Ø,38,C2,2Ø,E9,C1,CA,DØ,Ø47E
C45Ø:FA,6Ø,48,A5,FD,C5,FB,A5,Ø5A9
C458:FE,E5,FC,9Ø,Ø4,A5,A4,FØ,Ø5AC
C46Ø:Ø3,4C,7F,CØ,68,6Ø,AØ,ØØ,Ø2F6
C468:A2,ØØ,BØ,12,2Ø,52,C4,A1,Ø33B
C47Ø:FB,91,5C,2Ø,E9,C1,C8,DØ,Ø54A
C478:F3,E6,5D,18,9Ø,EE,2Ø,52,Ø43E
C48Ø:C4,B1,FD,81,5C,A5,FD,DØ,Ø5C1
C488:Ø2,C6,FE,C6,FD,A5,5C,DØ,Ø55A
C49Ø:Ø2,C6,5D,C6,5C,18,9Ø,E6,Ø3D5
C498:2Ø,ØE,C1,2Ø,ØE,C1,85,B4,Ø317
C4AØ:A2,ØØ,86,FB,86,FC,C9,24,Ø492
C4A8:FØ,55,C9,25,FØ,3E,C9,21,Ø44B
C4BØ:DØ,5Ø,2Ø,ØE,C1,C9,3A,BØ,Ø3C2
C4B8:2B,E9,2F,9Ø,25,A8,Ø6,FB,Ø3A1
C4CØ:26,FC,A5,FB,A6,FC,ØA,26,Ø494
C4C8:FC,ØA,26,FC,18,65,FB,9Ø,Ø43Ø
C4DØ:Ø1,E8,85,FB,98,18,65,FB,Ø479
C4D8:85,FB,8A,65,FC,85,FC,4C,Ø538
C4EØ:B2,C4,69,3Ø,6Ø,2Ø,ØE,C1,Ø35E
C4E8:C9,2Ø,DØ,Ø3,2Ø,ØE,C1,4A,Ø2F5
C4FØ:49,18,DØ,Ø7,26,FB,26,FC,Ø37B
C4F8:4C,E5,C4,49,18,2A,6Ø,2Ø,Ø3ØØ
C5ØØ:ØE,C1,AA,C9,47,BØ,1E,E9,Ø44Ø
C5Ø8:2F,9Ø,1A,C9,ØA,9Ø,Ø6,C9,Ø3ØB
C51Ø:11,9Ø,12,E9,Ø7,A2,Ø3,26,Ø26E
C518:FB,26,FC,CA,1Ø,F9,Ø5,FB,Ø4FØ
C52Ø:85,FB,4C,FF,C4,8A,6Ø,A9,Ø522
C528:ØØ,A2,Ø2,9D,ØA,Ø2,CA,1Ø,Ø227
C53Ø:FA,A2,ØF,Ø6,FB,26,FC,F8,Ø4C6
C538:AØ,Ø2,B9,ØA,Ø2,79,ØA,Ø2,Ø1EC
C54Ø:99,ØA,Ø2,88,1Ø,F4,CA,1Ø,Ø3ØB
C548:EA,D8,A2,Ø5,2Ø,64,C5,Ø9,Ø3BB
C55Ø:3Ø,9D,ØØ,Ø1,CA,1Ø,F5,A2,Ø33F
C558:Ø4,BD,ØØ,Ø1,C9,3Ø,DØ,Ø3,Ø28E
C56Ø:CA,DØ,F6,6Ø,A9,1Ø,ØE,ØC,Ø3C3
C568:Ø2,2E,ØB,Ø2,2E,ØA,Ø2,2A,ØØA1
C57Ø:9Ø,F4,6Ø,2Ø,E1,FF,EA,DØ,Ø59E
C578:F9,4C,7F,CØ,A9,2Ø,AØ,27,Ø414
C58Ø:91,D1,88,1Ø,FB,C8,84,D3,Ø514
C588:A9,2C,2Ø,FA,C1,4C,61,C2,Ø41F
C59Ø:A2,Ø3,B4,FB,B5,5C,94,5C,Ø455
C598:95,FB,CA,1Ø,F5,6Ø,C4,5C,Ø4DF
C5A0:8A,E5,5D,9Ø,ØB,18,98,E5,Ø3FC
C5A8:5E,8A,E5,5F,BØ,Ø2,18,6Ø,Ø356
C5BØ:38,6Ø,98,48,8A,48,A4,FB,Ø3E9
C5B8:A6,FC,2Ø,9E,C5,68,AA,68,Ø49F
C5CØ:A8,6Ø,98,38,E5,FB,48,8A,Ø48A
C5C8:E5,FC,A8,18,68,69,7E,BØ,Ø4AØ
C5DØ:Ø1,88,C8,DØ,Ø6,38,E9,8Ø,Ø3C8
C5D8:18,C8,6Ø,A2,55,2Ø,A2,C3,Ø3BC
C5EØ:A2,29,2Ø,AF,C3,2Ø,92,C1,Ø3DØ
C5E8:38,6Ø,18,98,65,57,A8,8A,Ø336
C5FØ:65,58,AA,6Ø,18,C8,6Ø,2Ø,Ø327
C5F8:27,C5,EA,BD,ØØ,Ø1,2Ø,D2,Ø386
C6ØØ:FF,CA,1Ø,F7,6Ø,2Ø,ØE,C1,Ø41F
C6Ø8:A2,Ø7,2Ø,ØE,C1,4A,49,18,Ø243
C61Ø:DØ,Ø9,26,B5,CA,1Ø,F3,A5,Ø426
C618:B5,38,6Ø,18,6Ø,2Ø,ØE,C1,Ø2B4
C62Ø:C9,3Ø,9Ø,F7,C9,47,BØ,F3,Ø533
C628:C9,3A,9Ø,ED,C9,41,BØ,E9,Ø523
C63Ø:6Ø,EA,A5,FD,C5,FB,A5,FE,Ø64F
C638:E5,FC,9Ø,F4,AØ,ØØ,B1,FB,Ø5B1
C64Ø:2Ø,38,C2,BØ,6E,C9,ØC,FØ,Ø3FD
C648:42,C9,Ø3,8A,48,C8,B1,FB,Ø454
C65Ø:48,CA,CA,FØ,Ø4,C8,B1,FB,Ø544
C658:AA,68,A8,68,9Ø,17,48,2Ø,Ø331
C66Ø:9E,C5,68,BØ,1Ø,C9,Ø2,FØ,Ø446
```

```
C668:11,20,EA,C5,98,A0,01,91,03AA
C670:FB,C8,8A,91,FB,20,44,C4,0501
C678:F0,B8,20,EA,C5,D0,05,98,04E4
C680:A0,01,D0,EF,A2,5B,20,DD,045A
C688:C5,B0,EA,20,67,C3,20,9E,0467
C690:C5,B0,0B,20,B2,C5,90,DD,0484
C698:20,EA,C5,18,90,0E,20,B2,0357
C6A0:C5,B0,D2,38,98,E5,57,A8,04FB
C6A8:8A,E5,58,AA,20,C2,C5,90,04A8
C6B0:C2,B0,C2,A2,41,20,A2,C3,049C
C6B8:A2,29,20,AF,C3,20,92,C1,03D0
C6C0:60,A0,00,20,0E,C1,90,2A,02A9
C6C8:C9,20,F0,F7,C9,22,F0,0F,04BA
C6D0:20,20,C6,20,27,C1,90,17,02B5
C6D8:91,FB,20,E9,C1,D0,E4,20,052A
C6E0:0E,C1,90,0E,C9,22,F0,DB,0423
C6E8:91,FB,20,E9,C1,D0,F0,4C,0562
C6F0:E5,C3,60,EA,EA,20,0E,C1,04CB
C6F8:20,47,C1,20,C1,C6,A0,57,03C6
C700:20,F4,C1,20,92,C1,4C,7F,0413
C708:C0,20,6D,C1,90,03,20,D4,0395
C710:C3,90,0E,A5,FD,85,5E,A4,048A
C718:FE,84,5F,20,0E,C1,20,1D,030D
C720:C1,90,76,8D,0A,02,85,57,033C
C728:46,B4,B0,33,A2,00,86,58,035D
C730:20,32,C6,A2,00,AC,0A,02,0272
C738:A5,FD,C5,5C,A5,FE,E5,5D,05A8
C740:90,0F,A1,FD,91,FD,A5,FD,056D
C748:D0,02,C6,FE,C6,FD,18,90,0501
C750:E7,A9,EA,C0,00,F0,05,91,04C0
C758:FD,88,D0,FB,4C,7F,C0,A9,0584
C760:00,E5,57,85,57,A9,00,E9,03AA
C768:00,85,58,20,32,C6,AC,0A,02AB
C770:02,A2,00,A5,FD,C5,5C,A5,040C
C778:FE,E5,5D,90,0C,B1,5C,81,046A
C780:5C,E6,5C,D0,02,E6,5D,D0,0483
C788:EA,38,A5,FD,ED,0A,02,85,0442
C790:FD,B0,02,C6,FE,A9,00,F0,050C
C798:BA,4C,E5,C3,20,6D,C1,90,048C
C7A0:F8,20,90,C5,20,6D,C1,90,044B
C7A8:F0,20,90,C5,20,1D,C1,90,03F3
C7B0:E8,48,20,24,C1,90,E2,48,03EF
C7B8:20,0E,C1,A2,00,C9,2D,F0,0377
C7C0:04,A2,02,C9,2B,F0,03,4C,02DB
C7C8:CD,C3,68,F5,5C,85,57,68,048D
C7D0:F5,5D,85,58,20,32,C6,4C,0393
C7D8:7F,C0,20,6D,C1,20,90,C5,0402
C7E0:A2,FF,86,FD,86,FE,A6,2B,0579
C7E8:A5,2C,86,2D,85,2E,A0,01,02D8
C7F0:B1,2D,F0,26,C8,B1,2D,85,041F
C7F8:FD,C8,B1,2D,85,FE,C8,F0,05DE
C800:04,B1,2D,D0,F9,38,98,65,03E0
C808:2D,AA,A5,2E,69,00,48,A0,02FB
C810:01,91,2D,88,8A,91,2D,68,02F7
C818:D0,D0,A5,37,C5,2D,A5,38,044B
C820:E5,2E,90,7C,F0,7A,E6,FD,056C
C828:D0,02,E6,FE,A5,FE,C9,FA,061C
C830:B0,71,A5,5E,C5,5C,A5,5F,0449
C838:E5,5D,90,7B,A0,02,A5,FD,0491
C840:91,2D,C8,A5,FE,91,2D,C8,04AF
C848:A9,83,EA,EA,EA,EA,EA,2C,05EA
C850:A9,2C,91,2D,C8,84,B4,A0,0433
C858:00,B1,5C,85,FB,84,FC,20,042D
C860:27,C5,EA,EA,A4,B4,BD,00,04D5
C868:01,91,2D,C8,CA,10,F7,E6,043E
C870:5C,D0,02,E6,5D,C0,45,B0,0426
C878:0A,A5,5E,C5,5C,A5,5F,E5,0417
C880:5D,B0,CD,A9,00,91,2D,38,0379
C888:98,65,2D,48,A5,2E,69,00,02AE
C890:A0,01,91,2D,AA,68,88,91,038A
C898:2D,85,2D,86,2E,4C,1A,C8,02C1
C8A0:A2,64,2C,A2,71,20,A2,C3,03CA
C8A8:A2,29,20,AF,C3,A5,5D,20,037F
C8B0:9D,C1,A5,5C,20,9D,C1,A9,0486
C8B8:00,A8,91,2D,C8,91,2D,A5,0391
C8C0:2D,A6,2E,18,69,02,90,01,0215
C8C8:E8,85,2D,86,2E,85,2F,86,0388
C8D0:30,85,31,86,32,A5,37,A6,0320
C8D8:38,85,33,86,34,A2,19,86,02EB
C8E0:16,4C,7F,C0,A2,00,20,AF,0312
C8E8:C3,A0,3B,20,F4,C1,AD,00,0420
C8F0:02,20,9D,C1,AD,01,02,20,0250
C8F8:9D,C1,20,E1,C1,AD,07,02,03D6
C900:20,9D,C1,AD,08,02,20,9D,02F2
C908:C1,AD,02,02,20,23,C2,20,0297
C910:18,C2,20,CA,C1,4C,7F,C0,0410
C918:20,6D,C1,20,52,C4,20,73,0317
C920:C5,A0,3A,20,F4,C1,20,92,0426
C928:C1,A9,08,A6,13,F0,02,A9,03C6
C930:10,20,CA,C1,F0,E5,20,5A,040A
C938:C1,20,47,C1,90,36,20,BF,038E
C940:C1,20,44,C1,90,2E,A5,FB,0444
C948:8D,08,02,A5,FC,8D,07,02,02CE
C950:20,05,C6,90,1F,8D,02,02,022B
C958:20,18,C2,D0,0A,20,5A,C1,030F
C960:20,47,C1,90,0F,A9,08,85,02FD
C968:B5,20,0E,C1,20,02,C2,D0,0358
C970:F8,4C,7F,C0,4C,E5,C3,20,0497
C978:6D,C1,B0,0D,A9,FF,85,FD,0515
C980:85,FE,D0,05,20,E4,FF,F0,054B
C988:FB,A9,05,85,B4,A5,13,D0,046A
C990:04,C6,B4,30,EF,20,73,C5,03F5
C998:20,52,C4,20,5C,C2,20,44,02D8
C9A0:C4,F0,EA,20,44,C1,90,2D,0480
C9A8:20,0E,C1,20,1D,C1,90,12,028F
C9B0:A8,20,38,C2,98,A0,00,91,038B
C9B8:FB,C8,CA,F0,05,20,1D,C1,0480
C9C0:B0,F5,20,7C,C5,20,E4,C1,04CB
C9C8:20,44,C4,20,7C,C5,A2,00,032B
C9D0:86,D3,4C,D7,CD,4C,E5,C3,053D
C9D8:20,6D,C1,90,F8,20,0E,C1,03C5
C9E0:20,1D,C1,90,F0,A2,00,20,0340
C9E8:52,C4,81,FB,20,E9,C1,18,0474
C9F0:90,F5,20,EC,C3,4C,66,C4,04CA
```

```
C9F8:20,98,C4,A9,24,C5,B4,F0,04B2
CA00:06,20,2A,CA,20,92,C1,A9,0336
CA08:25,C5,B4,F0,0D,20,2A,CA,03AF
CA10:A5,FC,20,28,C2,A5,FB,20,046B
CA18:23,C2,A9,21,C5,B4,F0,06,041E
CA20:20,2A,CA,20,F7,C5,4C,7F,03BB
CA28:C0,EA,A2,3D,4C,AC,C1,20,0462
CA30:44,C1,90,03,20,BF,C1,AE,03E6
CA38:06,02,9A,78,AD,07,02,8D,025D
CA40:15,03,AD,08,02,8D,14,03,0173
CA48:AD,00,02,48,AD,01,02,48,01EF
CA50:AD,02,02,48,AD,03,02,AE,0259
CA58:04,02,AC,05,02,40,AE,06,01AD
CA60:02,9A,A2,80,6C,00,03,20,024D
CA68:6D,C1,90,6B,20,90,C5,20,03BE
CA70:6D,C1,20,90,C5,20,52,C4,03D9
CA78:20,73,C5,20,81,C3,90,13,035F
CA80:20,9E,C5,B0,0E,A2,4D,20,0350
CA88:AF,C3,20,92,C1,20,DE,C1,04A4
CA90:20,88,C2,20,44,C4,F0,DD,045F
CA98:20,6D,C1,20,90,C5,A2,0B,0370
CAA0:A9,02,86,FB,85,FC,20,C1,048E
CAA8:C6,A5,FB,18,E9,0B,90,27,0429
CAB0:8D,0A,02,20,90,C5,20,52,0280
CAB8:C4,20,73,C5,AC,0A,02,B1,0385
CAC0:FB,D9,0B,02,D0,0B,88,10,0354
CAC8:F6,A2,4D,20,AF,C3,20,92,0429
CAD0:C1,20,E9,C1,18,90,DF,4C,045E
CAD8:E5,C3,20,6D,C1,A2,20,A9,0461
CAE0:3A,20,AC,C1,A4,FD,A6,FE,050C
CAE8:20,C2,C5,B0,03,20,9D,C1,03D8
CAF0:4C,7F,C0,4C,E5,C3,20,98,0437
CAF8:C4,C9,2C,D0,F6,20,7D,C1,04DD
CB00:20,9B,C4,C9,2C,D0,03,20,0367
CB08:0E,C1,A2,01,C9,41,D0,0B,0357
CB10:B5,FB,35,FD,95,FB,CA,10,054C
CB18:F7,30,0D,C9,45,D0,0B,B5,03D2
CB20:FB,55,FD,95,FB,CA,10,F7,05AE
CB28:30,0D,C9,4F,D0,0C,B5,FB,03E1
CB30:15,FD,95,FB,CA,10,F7,4C,04BF
CB38:B9,CB,C9,2B,D0,0D,18,A5,0412
CB40:FB,65,FD,AA,A5,FC,65,FE,060B
CB48:18,90,6A,C9,2D,D0,0C,A5,0389
CB50:FD,E5,FB,AA,A5,FE,E5,FC,070B
CB58:18,90,5A,A2,00,86,5C,86,030C
CB60:5D,C9,2A,D0,20,A5,FB,05,03E5
CB68:FC,F0,46,46,FC,66,FB,90,0565
CB70:0D,18,A5,5C,65,FD,85,5C,0369
CB78:A5,5D,65,FE,85,5D,06,FD,044A
CB80:26,FE,18,90,E0,C9,2F,D0,0474
CB88:3D,A5,FB,05,FC,F0,37,CA,04CF
CB90:06,FB,26,FC,90,F9,66,FC,050E
CB98:66,FB,38,A5,FD,E5,FB,A8,05C3
CBA0:A5,FE,E5,FC,90,04,84,FD,0599
CBA8:85,FE,26,5C,26,5D,E8,30,03A0
CBB0:E5,A6,5C,A5,5D,86,FB,85,04EF
CBB8:FC,A2,3D,A9,24,20,AC,C1,0435
CBC0:20,92,C1,4C,7F,C0,4C,E5,042F
CBC8:C3,A5,9D,09,40,85,9D,EA,045A
CBD0:EA,A0,00,20,0E,C1,90,1B,0324
CBD8:C9,20,F0,F7,C9,22,D0,E6,0571
CBE0:20,0E,C1,90,0E,C9,22,F0,0368
CBE8:0A,99,0A,02,C8,C0,11,90,02D8
CBF0:EF,B0,D3,98,AE,EA,CB,AC,0619
CBF8:EB,CB,20,BD,FF,EA,EA,A2,0608
CC00:01,20,0E,C1,90,14,C9,2C,0289
CC08:D0,BC,20,24,C1,90,B7,EA,04C2
CC10:EA,29,0F,F0,B1,C9,03,F0,047F
CC18:AD,AA,A0,01,20,BA,FF,EA,04BB
CC20:A6,B4,E0,0E,F0,16,20,0E,037C
CC28:C1,B0,9B,A9,00,20,D5,FF,04A9
CC30:20,B7,FF,29,10,F0,8C,A2,042D
CC38:84,4C,CF,C3,20,0E,C1,90,03E1
CC40:CC,C9,2C,D0,C3,20,47,C1,047C
CC48:90,F5,20,7D,C1,20,0E,C1,03D2
CC50:90,F6,C9,2C,D0,ED,20,47,049F
CC58:C1,90,F5,20,0E,C1,B0,C9,04AE
CC60:20,E4,C1,A9,FD,EA,A6,FB,05F6
CC68:A4,FC,20,D8,FF,4C,7F,C0,0522
CC70:4C,E5,C3,20.0E,C1,90,F8,046B
CC78:C9,20,F0,F7,20,2B,C1,20,03FC
CC80:4A,C1,90,EC,20,0E,C1,B0,0426
CC88:09,20,E1,C1,20,88,C2,4C,0381
CC90:D7,CD,C9,20,F0,EE,8D,0C,0504
CC98:02,20,0E,C1,8D,0B,02,20,01AB
CCA0:0E,C1,8D,0A,02,A9,39,85,02CF
CCA8:B5,C6,B5,30,C3,A2,02,A5,046C
CCB0:B5,0A,65,B5,A8,BD,0A,02,034A
CCB8:D9,18,CF,D0,EC,C8,CA,10,051E
CCC0:F4,A6,B5,8E,0A,02,A2,04,038F
CCC8:A9,20,9D,0B,02,CA,10,FA,0347
CCD0:A2,04,20,0E,C1,90,58,C9,0346
CCD8:20,F0,F7,C9,23,D0,31,20,0414
CCE0:1D,C6,B0,11,C9,27,D0,06,036A
CCE8:20,0E,C1,18,90,17,C9,24,029B
CCF0:D0,0A,20,1D,C6,20,29,C1,02E7
CCF8:B0,0B,90,07,C9,25,D0,00,0310
CD00:20,08,C6,90,6A,85,FD,A2,040C
CD08:02,86,B4,D0,41,20,0E,C1,033C
CD10:C9,24,F0,F9,9D,0B,02,20,03A0
CD18:20,C6,90,0D,20,27,C1,90,031B
CD20:0E,A4,FD,84,FE,85,FD,A9,055C
CD28:00,9D,0B,02,CA,10,DE,A9,030B
CD30:0C,85,B4,C6,B4,30,51,A5,03E5
CD38:B4,0A,0A,65,B4,69,04,A8,02F6
CD40:A2,04,BD,0B,02,D9,EC,CD,0402
CD48:D0,E9,88,CA,10,F4,A0,00,04AF
CD50:B9,E8,CF,29,0F,C5,B4,F0,0511
CD58:32,C8,C0,14,90,F2,A5,B4,04A9
CD60:D0,04,E6,B4,D0,E8,C9,07,04F6
CD68:D0,03,18,90,15,C9,09,D0,0332
CD70:17,AD,0A,02,C9,08,B0,10,0261
CD78:EA,A4,FD,A6,FE,20,C2,C5,05D6
CD80:85,FD,A9,0C,85,B4,90,C6,04C6
```

```
CD88:4C,E5,C3,AE,0A,02,E0,20,03AE
CD90:90,04,BD,E0,CE,2C,A9,FF,04D3
CD98:49,FF,39,C0,CF,1D,C8,CE,04C3
CDA0:20,38,C2,C5,B4,D0,B2,CA,04DF
CDA8:F0,0F,CA,F0,06,A0,02,A5,0406
CDB0:FE,91,FB,A0,01,A5,FD,91,055E
CDB8:FB,A0,00,A5,B5,91,FB,20,04A1
CDC0:38,C2,8A,18,65,FB,85,FB,047C
CDC8:90,02,E6,FC,20,E4,C1,A9,04E2
CDD0:41,20,FA,C1,20,92,C1,A9,0438
CDD8:1D,A2,0A,86,C6,9D,76,02,032A
CDE0:CA,D0,FA,86,D3,4C,8E,C0,0587
CDE8:FF,FF,FF,FF,20,20,20,20,047C
CDF0:20,20,20,20,20,41,20,20,0121
CDF8:20,00,23,20,20,58,2C,00,0107
CE00:20,20,59,2C,00,29,58,2C,0172
CE08:00,28,59,2C,29,00,28,20,011E
CE10:20,20,20,00,20,29,00,00,00A9
CE18:28,20,20,20,00,00,20,58,0100
CE20:2C,00,00,20,59,2C,00,00,00D1
CE28:0D,20,20,20,20,20,50,43,0140
CE30:20,20,49,52,51,20,20,4E,01BA
CE38:56,2D,42,44,49,5A,43,20,020F
CE40:41,43,20,58,52,20,59,52,0219
CE48:20,53,D0,20,45,52,52,4F,029B
CE50:D2,20,49,4E,A0,4D,49,53,0312
CE58:53,49,4E,47,20,2B,2F,AD,0258
CE60:4C,4F,57,20,53,54,41,43,023D
CE68:CB,49,4C,4C,45,47,41,4C,02C5
CE70:20,43,4F,44,C5,0D,2E,46,023C
CE78:4F,55,4E,44,A0,42,52,41,02AB
CE80:4E,43,C8,5A,45,52,4F,20,02B9
CE88:50,41,47,C5,4F,55,54,20,02B5
CE90:4F,46,20,4D,45,4D,4F,52,0235
CE98:D9,49,4C,4C,45,47,41,4C,02D3
CEA0:20,4C,49,4E,45,20,4E,55,020B
CEA8:4D,42,45,D2,4C,4F,41,C4,0346
CEB0:52,4D,3B,3A,44,2C,46,54,021E
CEB8:43,48,47,58,42,4C,53,4F,025A
CEC0:41,4B,49,57,52,48,44,00,020A
CEC8:90,B0,F0,30,D0,10,50,70,0400
CED0:00,18,D8,58,B8,CA,88,E8,043A
CED8:C8,EA,48,08,68,28,40,60,0332
CEE0:AA,A8,BA,8A,9A,98,38,F8,04F8
CEE8:41,81,E1,22,01,42,A0,A2,034A
CEF0:A1,C1,02,21,61,4C,84,86,033C
CEF8:62,E6,C6,E0,C0,24,20,78,046A
CF00:E3,E3,E3,E3,E3,E3,E3,E3,0718
CF08:E3,E3,E3,E3,E3,DF,E7,E7,071C
CF10:E3,E7,E7,F3,F3,F7,FF,FF,078C
CF18:42,43,43,42,43,53,42,45,0227
CF20:51,42,4D,49,42,4E,45,42,0240
CF28:50,4C,42,56,43,42,56,53,0262
CF30:42,52,4B,43,4C,43,43,4C,0240
CF38:44,43,4C,49,43,4C,56,44,0245
CF40:45,58,44,45,59,49,4E,58,026E
CF48:49,4E,59,4E,4F,50,50,48,0275
CF50:41,50,48,50,50,4C,41,50,0256
CF58:4C,50,52,54,49,52,54,53,0284
CF60:54,41,58,54,41,59,54,53,0282
CF68:58,54,58,41,54,58,53,54,0298
CF70:59,41,53,45,43,53,45,44,0251
CF78:45,4F,52,53,54,41,53,42,0263
CF80:43,52,4F,4C,4F,52,41,4C,025E
CF88:53,52,4C,44,59,4C,44,58,0276
CF90:4C,44,41,43,4D,50,41,53,0245
CF98:4C,41,4E,44,41,44,43,4A,0231
CFA0:4D,50,53,54,59,53,54,58,029C
CFA8:52,4F,52,49,4E,43,44,45,0256
CFB0:43,43,50,58,43,50,59,42,025C
CFB8:49,54,4A,53,52,53,45,49,026D
CFC0:6C,9E,20,A2,96,0A,8A,80,0376
CFC8:09,19,10,01,11,08,03,04,0053
CFD0:00,0C,1C,14,FF,DF,FF,FF,0418
CFD8:DF,9F,8F,9F,1F,1F,1F,1F,0328
CFE0:1F,0F,03,1C,9F,1C,1C,1C,0140
CFE8:38,3B,39,22,24,11,10,22,0135
CFF0:22,3B,2C,25,26,10,1F,27,012A
CFF8:10,39,3A,23,00,52,41,FF,0238
D000:00,00,00,00,00,00,00,00,0000
```

Macro Assembler/Editor

Lunghezza del programma: 9910 byte. Linguaggio di programmazione: Assembler. Accessori: drive e stampante.

Chi vuole sfruttare appieno le possibilità del suo C64, prima o poi si accorgerà di problemi risolvibili soltanto con la formulazione di segmenti di programma nell'Assembler 6502. I due problemi più ricorrenti sono la scarsa velocità o la capacità limitata dell'area RAM in BASIC.
Anche se il Commodore BASIC supporta l'inserimento di routine in codice macchina in programmi BASIC, mediante USR, SYS, PEEK e POKE, nella versione di base non c'è la possibilità di scrivere programmi di questo genere.

Perciò si rendono necessari programmi in Assembler o programmi monitor, che aiutano l'utente nella stesura di programmi in codice macchina. I programmi introdotti sotto monitor vengono memorizzati direttamente come codice macchina eseguibile. Tutti gli indirizzi però devono già essere determinati in fase di input e le modifiche risultano piuttosto dispendiose. Un monitor è adatto alla stesura di brevi programmi ed è uno strumento irrinunciabile per l'analisi e la correzione di programmi scritti da altri. Per questo motivo in questo fascicolo abbiamo pubblicato un monitor, che può lavorare con questo Assembler. Programmando in Assembler, innanzitutto viene generato un file fonte sotto editor, che poi viene convertito nel codice macchina eseguibile. Nel testo fonte al posto delle costanti possono comparire le cosiddette label, a cui vengono assegnati dei valori. Se per esempio si utilizza un sottoprogramma per l'output di un carattere su video, si può definire una label con "BSOUT = $FFD2" e al posto di 'JSR $FFD2' si può usare la forma 'JRS BSOUT', molto più trasparente. È possibile anche inserire dei commenti nel testo del programma. Un programma fonte così generato e commentato è molto più facile da leggere e da capire del programma in codice macchina generato sotto monitor. Chi ha provato a modificare un programma dopo alcune settimane o alcuni mesi dalla prima stesura, capisce i vantaggi di un listato commentato. Se per esempio l'indirizzo iniziale del programma generato deve venire cambiato, basta modificare solo l'assegnazione dell'indirizzo iniziale nel testo fonte; tutti gli indirizzi interni vengono ricalcolati automaticamente dall'Assembler.
Il programma in codice macchina presentato in questo articolo, della lunghezza di 9,5 Kb, comprende un Assembler e un editor per la generazione di testi fonte. L'unione di Assembler ed editor rende particolarmente confortevole la stesura dei programmi, in quanto gli errori segnalati dall'Assembler possono venire eliminati immediatamente dall'editor. L'area di lavoro specificata nella definizione delle label e nel testo fonte può venire scelta a piacere. Nel caso di programmi molto lunghi l'Assembler offre la possibilità di dividere l'area di lavoro in più moduli, che vengono gestiti da un modulo di controllo. Il programma in codice macchina generato può venire registrato direttamente in memoria o su disco come file di programma.
Il programma lavora indipendentemente dalle ROM BASIC, che possono venire disattivate per utilizzare l'area libera, per esempio per la memorizzazione del testo fonte. Tuttavia bisogna fare attenzione che nel caso di partenza a freddo il contenuto della cella $A000 viene modificato dalla routine di memorizzazione del sistema operativo.
Il programma supporta tutte le periferiche collegate al bus seriale IEC; tuttavia non abbiamo considerato il collegamento di un registratore in considerazione della lunghezza del programma. L'output dell'Assembler su stampante avviene mediante l'unità #4, indirizzo secondario #0; questa regolazione si può modificare con dei POKE.
Una volta che l'Assembler è stato caricato, può essere lanciato "a freddo" da BASIC con 'SYS31232', ma in questo caso il testo fonte eventualmente in memoria viene cancellato (si può recuperare con UNNEW); con 'SYS31235' invece si lancia l'Assembler senza perdere il testo fonte.
Chi ha già introdotto il programma Monitor, con:
.T 7A00,9FFF,0A00—
.ER 0A00,2FFF,7A00,9FFF,0A00—
può adattare l'Assembler all'area di memoria $0A00-$3000; l'errore che viene segnalato non ha significato.
Con i seguenti POKE si può modificare la regolazione del canale di stampa e il formato delle pagine.

1.) Canale di stampa:
POKE 34771, indirizzo della stampante (4)
POKE 34773 indirizzo secondario (0)
2.) Formato delle pagine:
POKE 34307,linee utilizz./pag. (63)
POKE 34407, linee vuote tra le pagine (9)
POKE 34312, ampiezza delle linee+1:
POKE 34417,ampiezza delle linee+1 (80)
POKE 34321,posizione del numero della pagina (32)

L'editor

L'editor lavora per linee, come l'editor BASIC, tuttavia è abbastanza veloce da poter essere usato per testi fonte molto lunghi. Tutti gli output dell'editor possono essere inviati alla stampante facendo precedere il comando da una 'P'; inoltre esiste la possibilità di suddividere l'output su stampante in pagine. Se una linea è introdotta da spazi vuoti, viene prodotto il loro numero, non dei blank. I comandi di editor possono venire abbreviati, come in BASIC, introducendo il secondo carattere e i successivi con SHIFT.
Tutti gli output, per esempio i listati, possono essere sospesi o resi di nuovo mediante il tasto '-' e terminati con RUN/STOP.

I comandi editor

I parametri tra parentesi [] non devono essere introdotti, saranno sostituiti in seguito da un valore sensato; se si introducono dei parametri, si devono eliminare le parentesi. Quando nella descrizione due caratteri sono separati da '/', si può introdurre a scelta il primo o il secondo. L'indicazione di un gruppo di linee ha lo stesso formato del BASIC:
LIST 10-999; dalla linea 10 alla 999
DELETE 10-; dalla linea 10 alla fine del programma
LIST -1000; dall'inizio del programma alla linea 1000.

AUTO [ampiezza del passo]
Dopo l'input di una linea viene prodotto automaticamente il numero di linea successivo, secondo il passo indicato. L'ampiezza del passo può essere un valore tra 1 e 255; se non viene indicata o se viene settata a 0, il modo AUTO viene disattivato. Viene prodotto un nuovo numero di linea quando il primo carattere dell'ultima linea introdotta è una cifra. Alla chiamata dell'Assembler l'ampiezza del passo è di 10.

RENUM [gruppo di linee],ampiezza del passo
Il gruppo di linee viene rinumerato secondo il passo indicato. Se la rinumerazione non è possibile, per esempio perché il numero della linea successiva al gruppo è inferiore al numero dell'ultima linea del gruppo, viene visualizzato un ILLEGAL QUANTITY e il comando non viene eseguito. La prima linea del gruppo indicato conserva il numero della linea iniziale indicata nel blocco. Per i gruppi di linee con lo stesso numero (per esempio dopo MOVE o COPY) basta indicare questo numero.

MOVE [gruppo di linee], numero destinazione
Il gruppo di linee viene shiftato alla destinazione indicata e il gruppo originale viene cancellato. Tutte le linee della copia conservano il numero della linea destinazione e devono essere rinumerate con RENUM. MOVE sposta una linea per volta, perciò è possibile shiftare grossi gruppi anche se la memoria ancora libera è limitata; tuttavia l'esecuzione del comando è molto lenta. Per questo motivo in molti casi è preferibile un uso combinato di COPY e di DELETE.

COPY [gruppo di linee], numero destinazione
Il gruppo di linee indicato viene copiato al numero destinazione, mentre il gruppo originale resta intatto. Dal momento che tutte le linee hanno il numero destinazione, bisogna rinumerarle con RENUM. Il numero destinazione non deve essere contenuto nel gruppo da copiare.

DELETE [gruppo di linee]
Il gruppo indicato viene cancellato; se non si specifica nessun numero, il testo fonte viene cancellato completamente, ma può essere recuperato con UNNEW.

COLD
Esegue un avvio "a freddo" dell'Assembler e il testo fonte in memoria viene cancellato.

UNNEW
Serve a recuperare un testo fonte che si trovava in memoria ed è stato cancellato. UNNEW inoltre deve essere dato quando si hanno delle incertezze sul formato del testo fonte (per esempio se un programma in codice macchina può averlo modificato). Il formato viene controllato e il contatore del testo fonte viene resettato.

FIND "stringa da ricercare",[gruppo di linee]
La stringa da ricercare viene cercata nel gruppo di linee indicato e tutte le linee che la contengono vengono listate. Il carattere "\" nella stringa di ricerca può indicare un qualsiasi carattere, tranne lo spazio vuoto. Al posto delle virgolette come delimitatori di stringa si possono usare anche altri caratteri. Se una stessa linee contiene più occorrenze della stringa, viene listata comunque una sola volta. Dopo la ricerca viene indicato il numero delle occorrenze della stringa riscontrate. Per esempio:
FIND .abc. 10-10000;
cerca 'abc' nelle linee da 10 a 10000.
FIND ' "abc" '
cerca in tutto il testo la stringa 'abc' racchiusa tra '?'.

REPLACE "stringa da ricercare"stringa da sostituire" [*] [gruppo di linee]
La stringa ricercata nel gruppo di linee viene sostituita dalla seconda stringa indicata. Per quanto riguar-

da i delimitatori delle stringhe vale quanto abbiamo detto per FIND. Quando viene indicato '*', il programma si ferma ad ogni occorrenza della stringa e chiede di confermare la sostituzione con 'Y' oppure RETURN. Se viene battuto un altro tasto, la ricerca continua senza operare la sostituzione. Infine REPLACE indica il numero delle sostituzioni eseguite.

LIST [gruppo di linee]
Il gruppo di linee indicato viene listato.

FORMAT C/[#[,#]]
FORMAT C attiva la formattazione automatica e dà la possibilità di formattare i listati, cosa molto utile nel caso di programmi nidificati. Gli argomenti indicano l'ampiezza della finestra label e l'ampiezza della finestra comandi. Lo standard è 8 caratteri per le label e 20 caratteri per i comandi.

P comando
Tutti gli output vengono inviati alla stampante.
Per esempio: PLIST 10-999;
le linee 10-999 vengono stampate.
PASSEMBLE L;
il protocollo generato durante l'Assembler viene stampato.

PRINT [?testo?/espr.simb]
Il testo o il valore dell'espressione simbolica vengono prodotti. Se l'output di PRINT avviene su stampante, viene formattato da PAGE.

PAGE [+/numero pagina]
PAGE + aumenta di uno il numero di pagina corrente. Nel caso di output su stampante (PPAGE+), si ha un avanzamento alla pagina successiva. PAGE senza argomento disattiva il modo PAGE e l'output diventa continuo. PAGE con specifica di un numero attiva il modo PAGE e setta il numero di pagina al valore indicato.

LOAD [g#] ?nome?
Il testo fonte memorizzato sotto il nome indicato nel floppy numero g# viene caricato in memoria. Se g# non viene specificato, come valore di default si usa 8.

APPEND [g#] ?nome?
Funziona come LOAD, ma il nuovo testo viene aggiunto a quello già presente in memoria.

SAVE [g#] ?nome? [gruppo di linee]
Il gruppo di linee viene salvato sull'unità g# (valore di default 8) sotto il nome indicato.

DC [g#[[,s#] ?stringa?][;]]
La stringa viene inviata all'unità g#, indirizzo secondario s#. Se viene specificato ';', non viene inviato il carriage return finale. I valori di default sono 8 per g# e 15 per s#. Se s# e la stringa non vengono specificati, il floppy indirizzato segnala un errore.
Per esempio: DC 9:?S:PROGRAMMA?
il programma sul dischetto nel floppy 9 viene cancellato;
DC:9
si richiede un messaggio di errore dal floppy;
C 4,7 ??:
all'unità numero 4/indirizzo secondario 7 viene prodotto un CR.

DIR [g#]
La directory del floppy g# (valore di default 8) viene visualizzata.

BASIC
Si torna sotto BASIC.

BRK
Esecuzione del comando codice macchina BRK e salto al programma monitor.

MEMORY [indirizzo iniziale[-indirizzo finale]]
I confini dell'area di memoria vengono fissati ai due nuovi valori specificati. I valori possono essere decimali o esadecimali, con il prefisso '$'. Ricordate che la memoria di lavoro non può estendersi nella zero-page, nell'area del sistema operativo, di I/O e dell'Assembler. Il testo fonte contenuto in memoria viene shiftato automaticamente alla nuova area. MEMORY viene eseguito solo se la nuova area è abbastanza grande da contenere il testo corrente. Se non vengono specificati gli argomenti, vengono prodotti i confini dell'area attuale e la memoria ancora libera.
Per esempio:
MEMORY $1000-$7A00;
dice che l'area $1000-$7A00 è libera come memoria di lavoro.

Comandi in riferimento all'Assembler

ASSEMBLE [linea] [M/L]
Il testo fonte in memoria viene assemblato dalla linea indicata; in questo modo si possono avere in memoria piccoli moduli, assemblati separatamente. Indicando 'M' o 'L', il protocollo Assembler è prodotto con o senza codice oggetto macro.

LABEL
Le label "normali", definite durante l'assemblaggio, vengono prodotte in ordine di definizione. Gli indirizzi della zero-page vengono listati a parte.

SYS espr.simb
Un programma in codice macchina viene lanciato dall'indirizzo indicato.

L'Assembler

Si tratta di un Assembler a due passate, cioè che passa in rassegna il testo fonte due volte per tradurlo. Alla prima passata viene definata la tabella dei simboli e viene data una prima indicazione degli errori; alla seconda passata si esegue la traduzione vera e propria. Il programma può venire prodotto in memoria e/o come modulo di caricamento assoluto su disco. L'Assembler conosce tutti i comandi del 6502 con i relativi modi di indirizzamento e una serie di pseudocomandi; inoltre permette la definizione di macro. Le label possono contenere otto caratteri significativi; gli altri caratteri vengono ignorati. I messaggi di errore vengono visualizzati sotto forma di testo con il rispettivo codice di errore, per un riconoscimento immediato dell'errore.

Struttura del testo fonte

Una linea del testo fonte ha la seguente forma:
Num. di linea [label][comando [operando]][;commento]

Il numero di linea

Sono ammessi tutti i valori da 0 a 65535.

Simboli

Come primo carattere sono consentite le lettere da 'a' a 'z'; come caratteri successivi sono consentiti anche 0..9 . # " $ e \. Per evitare confusione non esiste distinzione tra maiuscole/minuscole, quindi 'LABEL' equivale a 'label'. Quando si definisce un simbolo, il primo carattere deve occupare la colonna 0, cioè seguire immediatamente il numero di linea. È possibile assegnare lo stesso valore a più simboli.

Comando

Può essere un qualsiasi mnemo/macro o pseudocomando valido. Il comando deve essere separato dal campo precedente da almeno uno spazio vuoto. Le pseudo operazioni possono iniziare anche alla colonna 0.

Operando

Segue l'operando, separato da almeno uno spazio vuoto. Può essere un'espressione simbolica (espr.simb) o una lista di parametri (nel caso di macro).

Commento

Si possono aggiungere dei commenti separati da ';'. Nelle linee che servono esclusivamente come commenti, ';' può trovarsi alla colonna 0.

Espr.simb

Le espressioni simboliche sono costanti o label, che vengono collegate da '+' o da '-'. Le espressioni non possono contenere spazi vuoti. Come costanti sono consentivi numeri decimali, esadecimali con prefisso '$' e binari con prefisso '%'. Lo stato attuale del contatore di programma (indica il codice del comando corrente) è indicato da '*'. Tutti i calcoli sono eseguiti con 16 bit interi di precisione, perciò l'espressione 32768+$8000 dà come risultato 0. I prefissi '<'e '>' selezionano rispettivamente il byte di valore minore e maggiore dell'espressione.

Rappresentazione dei modi di indirizzamento

Diretto

Esempi:

```
LDA #65           ;decimale
LDA #'A           ;ASCII CBM
LDA #?A           ;ASCII CBM
LDA #$41          ;esadecimale
LDA #%01000001    ;binario
LDA #LABEL        ;simbolico (il valore deve essere rappresentabile in 8 bit)
LDA #<espr.simb   ;seleziona il byte di valore minore
LDA #>espr.simb   ;seleziona il byte di valore maggiore
```

Assoluto/zero page

Esempio:

```
LDA espr.simb
```

Indic. assoluto/indic. zero-page

Esempi:

```
LDA espr.simb,X
LDA espr.simb,Y
```

Indiciamento indiretto

Esempio:

```
LDA (espr.simb,X)
```

Indiretto indiciato

Esempio:

```
LDA (espr.simb),Y
```

Indiretto

Esempio:

```
JMP (espr.simb)
```

Accumulatore

Esempio:

```
ASL A oppure ASL
```

Implicito

Esempio:

```
CLC
```

Relativo

Esempio:
BNE espr.simb
speciale: BNE *+2 viene tradotto come 'D0 00'

La scelta della zero-page o dell'indirizzamento a 16 bit è automatica. Quando il valore di un'espressione non è noto, si sceglie l'indirizzamento a 16 bit. Tuttavia si può ottenere l'indirizzamento zero-page con il prefisso '<'. Per esempio:

```
0073- EE 00 7B 00010 CHRGET INC CG3+1
0076- D0   02  00020 BNE CG3
0078- E6   7C  00030 INC<CG+2
007A- AD FF FF 00040 CG3 LDA $FFFF
007D- 60       00050 RTS
```

Al primo INC è stato scelto l'indirizzamento a 16 bit, perché il valore di CG3 non era ancora noto. Al secondo INC è stato selezionato il byte di valore minore e così l'indirizzamento scelto è zero-page.

Pseudo operazioni

simbolo = espr.simb
Al simbolo viene assegnato il valore che si ricava dall'espressione. Se il simbolo è già stato definito, compare il messaggio REDEF'D LABEL e l'assegnazione non ha luogo.

simbolo ← espr.simb
Come '='nel caso di un simbolo già esistente non viene segnalato un errore e si ha l'assegnazione.

* = espr.simb
Il contatore di programma viene settato al valore ricavato dall'espressione simbolica. espr.simb non può contenere una label il cui valore viene definito più avanti, altrimenti si ha un ILLEGAL FWD. REFERENCE e l'assemblaggio si interrompe. Quando il programma viene salvato su disco, il nuovo PC non può essere inferiore al precedente. Nel caso di output su disco l'eventuale spazio vuoto viene riempito con $00.
Esempio:
* = $C000; PC viene settato a $C000.

.OBJ [espr.simb]
Il codice oggetto viene depositato a partire dall'indirizzo risultante dalla espr.simb. Il valore di tutti gli elementi dell'espressione deve essere già stato definito. Se manca la specifica dell'indirizzo, in memoria non viene depositato alcun codice.
Esempi:
.OBJ *
il codice oggetto viene depositato nell'area originaria;
.OBJ $A000
il codice viene memorizzato a partire da $A000;
.OBJ
in memoria non viene depositato alcun codice. Attenzione: quando si setta il contatore di programma, viene corretto anche il codice oggetto. .OBJ quindi dovrebbe stare immediatamente dopo la definizione dell'indirizzo di base.
Esempi:

```
00000 .OBJ $A000
00010 * = $1000
```

il codice viene memorizzata da $B000 e PC viene predefinito.

```
00000 * = $1000
00010 .OBJ $A000
```

il codice viene memorizzato da $A000.

.OUT [[g#] ?nome?]
Il codice generato viene memorizzato direttamente sul drive g# sotto il nome indicato. .OUT senza il nome chiude un file di output aperto. Nell'assegnazione del PC bisogna fare attenzione che il nuovo stato del PC sia superiore a quello precedente.

.LIST [M]
Viene prodotto un protocollo Assembler con o senza codice macro.

.NOLIST [M]
L'output del protocollo o del codice a macro viene interrotto.

.BYT
Valuta le espressioni simboliche o le stringhe e le memorizza come sequenza di byte. Quando si utilizzano delle espressioni simboliche, il loro valore deve essere rappresentabile con 8 bit. Gli elementi di .BYT devono essere separati l'uno dall'altro da almeno uno spazio vuoto. Come delimitatori si possono usare: . / # ! " ' ecc. Per esempio:
.BYT /stringa/ 13 $d %0 >LABEL
memorizza stringa, il decimale 13, l'esadecimale $0d, il binario %00000000 e il byte di LABEL di valore inferiore.

.DBYT
Valuta espressioni simboliche a 16 bit e memorizza per primo il byte di valore maggiore. I singoli elementi devono essere separati da spazi vuoti. Per esempio: 12 34 56 78.

.WORD
È uguale a .DBYT, ma è il byte di valore inferiore ad essere memorizzato per primo.

.DEF simbolo [(parametri)]
Definisce un comando macro.

.^!DEF
Conclude una definizione di macro.

.IF espr.simb relop espr.simb
Se la condizione è soddisfatta, il testo che segue viene assemblato. Le condizioni possibili sono: < >, =, <, =<, > e =>. Le espressioni devono essere separate da almeno uno spazio vuoto. Bisogna ricordare che vengono usati 16 bit interi di precisione senza segno, la condizione $0010—255 > $0010 perciò è soddisfatta! Le definizioni di macro sono per blocchi IF chiusi, quindi non si può entrarci od uscirne. Le condizioni IF vengono concluse da .END o da .^!DEF.

.^!IF
Chiude una condizione IF aperta.

.FILE [g#] ?nome?
Al posto di FILE viene assemblato il file indicato. FILE può trovarsi soltanto in modo comando.

.END
Segnala la fine del testo fonte. Normalmente .END si può tralasciare, ma i moduli di controllo devono essere sempre chiusi da .END altrimenti l'Assembler continua con l'ultimo segmento caricato.

.PRINT
Dopo PRT possono trovarsi espressioni o stringhe da produrre alla prima passata. In questo modo, durante l'assemblaggio, è possibile produrre lo stato dei contatori di programma più interessanti. Le espressioni il cui valore non è ancora noto, devono essere precedute da due '?'.

.PAGE
Nel caso di output su stampante se PAGE è attivo, la formattazione avviene con salto di pagina.

Assemblaggio condizionato

L'Assembler conosce la pseudooperazione .IF che permette di assemblare determinati segmenti di programma soltanto se è soddisfatta la condizione prestabilita. In questo modo è possibile anche definire dei comandi macro ricorsivi. I valori necessari a soddisfare la condizione devono essere definiti fin dalla prima passata, altrimenti l'Assembler non può calcolare quanta area di memoria riservare. Le condizioni

consentite sono: < >, =, <, = <, > e = > e la condizione deve essere separata dall'espressione da almeno uno spazio vuoto. Osservate che vengono utilizzati 16 bit di precisione senza segno. Se la condizione è soddisfatta, l'assemblaggio continua con la linea successiva. Altrimenti cerca l'indicatore di fine blocco, .^!IF o .END (.^!DEF), e prosegue da quel punto. È possibile nidificare 255 blocchi IF, con il relativo indicatore di fine. Le definizioni macro sono per blocchi IF chiusi, da cui non si può né uscire, né entrare.
Esempi:

```
000010FLAG = 1
00020BSOUT =$FFD2
 ...
01000 .IF FLAG < > 0
      la parte che segue verrà assemblata soltanto
01010 LDX #31
      se il valore di FLAG non è nullo;
01020L1 TXA
      Se FLAG = 0 l'assemblaggio
01030 EOR #95
      continua alla linea 1070.
01040 JRS BSOUT
01050 DEX
01060 BPL L1
01070 .^!IF
```

I comandi macro

Se in un programma ricorre spesso una sequenza di comandi, si può definire una macro. Ogni volta che l'Assembler trova questa macro, assemblerà la sequenza di comandi definita precedentemente. Le macro tuttavia non si devono confondere con i sottoprogrammi, perché vengono assemblate di nuovo ad ogni loro chiamata. L'Assembler offre inoltre la possibilità di inserire dei parametri nelle macro, perciò una sola definizione può servire per diverse sequenze simili di comandi. All'interno di una definizione si possono utilizzare soltanto label locali, contrassegnate dal prefisso '\'. A queste label si ha accesso soltanto nella macro a quel livello di esecuzione, quindi si evita l'errore 'REDEF'D LABEL'. Nel caso delle definizioni ricorsive bisogna fare particolare attenzione a definire meno label possibili, altrimenti la richiesta di spazio della tabella delle label aumenta troppo.

Parametri

Quando si devono trasmettere dei parametri, nella definizione della macro dopo la denominazione tra parentesi deve venire la cosiddetta lista dei parametri, che contiene le label, separate da spazi vuoti, a cui saranno assegnati i parametri. Se vengono utilizzate della label globali, ad ogni nuova chiamata vengono assegnati loro nuovi valori. Per evitare errori, queste label non dovrebbero essere usate anche nel programma principale. Al momento della chiamata alla denominazione della macro segue, se presente, la lista dei parametri tra parentesi. Qui possono comparire espressioni simboliche il cui valore sarà trasmesso ai parametri corrispondenti (call by value). Il numero dei parametri trasmessi naturalmente deve corrispondere a quelli richiesti nella definizione.

Area di validità

Una macro deve essere già stata definita al momento della sua prima chiamata. Le seguenti limitazioni valgono soltanto per un programma suddiviso in più parti che devono essere caricate una dopo l'altra dall'Assembler. Naturalmente si può accedere alla definizione soltanto finché essa si trova in memoria. Ne consegue che una macro che è stata definita in una parte da caricare in un secondo tempo è disponibile soltanto per quel modulo, fino alla chiamata .FILE successiva. Non appena viene caricato un nuovo modulo, l'Assembler ?dimentica? la definizione. Se però una macro si deve usare in tutto il corso del programma, esiste una soluzione: basta definire la macro nel modulo di controllo, che resta sempre in memoria. Una macro tuttavia non può contenere in nessun caso una chiamata .FILE.
Esempi:

```
00100.DEF PUSH
      Definizione di una macro che salva sul registro stack
00110 PHA
      il contenuto del registro.
00120 TXA
00130 PHA
00140 TYA
00150 PHA
00160.^!DEF
      La fine della definizione
00170;
00180 PUSH
      viene tradotta con la sequenza di comandi indicata.
```

Esempi:

```
00100.DEF INCX(DUMMY)
      Definizione di una macro con parametri
00110 .IF DUMMY>0
      definizione ricorsiva
00120 INX
00130 INCX (DUMMY—1)
00140.^!DEF
      La fine della definizione
00150;
00160 INCX(8)
      viene tradotta come 'E8 E8 E8 E8 E8 E8 E8 E8'.
```

Con l'aiuto dell'Assembler condizionato qui è stato definito un comando che aumenta il registro X del valore indicato. INCX chiama se stesso alla linea 130, ma con un parametro diminuito di 1. Non appena si arriva al valore 0, la condizione non è più soddisfatta e la ricorsività finisce.

Esempio:

```
00100.DEF PRINT (TESTO POS)
      Definizione con più parametri
00110 LDX #POS
00120\L1 LDA TESTO,X
      e label locali.
00130 BEQ \L2
00140 JRS BSOUT
00150 INX
00160 BNE \L1
00170\L2 .^!DEF
      Anche la fine della definizione può
00180;
      comprendere una label.
00190;
00200 PRINT (TESTO1 0)
      Il testo viene scritto dalla posizione 0.
00210 BRK
00220TESTO1 .BYT."HALLO" 0
0
```

Per \L1 e \L2 sono state utilizzate delle label locali per evitare un messaggio di errore alla seconda chiamata di PRINT. Al momento dell'assemblaggio \L2 indicherà l'indirizzo di BRK.

Concatenazione di più moduli

Il comando .FILE dell'Assembler permette di suddividere i programmi molto lunghi in moduli, chiamati e assemblati da un modulo di controllo. Il modulo di controllo resta in memoria durante tutto l'assemblaggio, perciò deve essere il più piccolo possibile. Normalmente il modulo di controllo contiene soltanto le definizioni delle macro globali e le chiamate di .FILE. .FILE è permesso soltanto nel modulo di controllo, quindi i moduli successivi non possono contenere chiamate. Il modulo di controllo deve essere

lanciato dall'Assembler direttamente dopo il caricamento dell'ultima parte. Le macro che vengono definite nei moduli caricati in seguito, sono valide soltanto all'interno di quei moduli, fino alla successiva chiamata di .FILE. Invece le macro definite nel modulo di controllo sono a disposizione di tutti i moduli. Il caricamento dei moduli ha l'inconveniente di richiedere grande spazio di memoria, ma se durante l'assemblaggio si verifica un errore, la parte in causa si trova in memoria e può essere immediatamente corretta. Prima di lanciare l'Assembler, bisogna cancellare la parte che non appartiene al modulo di controllo. Esempio:

```
00010 * = $A000
      Indirizzo di base $A000
00020 .OUT 9 ?PROGRAMMA?
      il codice va direttamente al floppy 9
00030 .LIST
      output del protocollo di assemblaggio senza codice macro
00040 .FILE ?LABEL?
      caricare LABEL dal floppy 8
00060 .FILE 8 ?MAIN?
      caricare MAIN dal floppy 8
00070 .FILE 9 ?SUB?
      caricare SUB dal floppy 9
00080 .END
      fatto.
```

ASSEMBLER

Durante la programmazione si è cercato di evitare il ricorso ad indirizzi dipendenti dal sistema. È stato usato l'indirizzo di partenza a caldo della ROM BASIC, indirizzato con JMP ($A002). Ad alcuni indirizzi di sistema non si è potuto rinunciare, si tratta precisamente di:
$0090 status di IEC
$00D0 contatore EOLN per l'input da video
$00D3 numero della colonna video corrente
$FF93 invia l'indirizzo secondario dopo LISTEN
$FF96 invia l'indirizzo secondario dopo TALK
$FF9F MATRIX controlla la matrice della tastiera
$FFA5 prende un byte dal bus seriale
$FFA8 trasmette un byte dall'accumulatore al bus seriale
$FFAB invia UNTALK
$FFAE invia UNLISTEN
$FFB1 invia LISTEN. Num. dell'unità nell'accumulatore
FFB invia TALK. Num. dell'unità nell'accumulatore
$FFCC CLRCH libera tutti i canali del sistema operativo file manager
SFFCF BASIN aspetta un input da tastiera (con cursore)
SFFD2 BSOUT produce su video il carattere nell'accumulatore
SFFE4 GET prende un carattere da tastiera
Come memoria di lavoro l'Assembler utilizza gli indirizzi da $0055 a $006B come memoria per i diversi indicatori, l'area $0200-$0258 come buffer di input e stack. Altri puntatori e registri si trovano alla fine del programma.

I messaggi di errore dell'Assembler

01 UNDEF'D LABEL
È stato usato come argomento un simbolo non definito. Questo messaggio compare soltanto alla seconda passata.
02 TYPE MISMATCH
È stata utilizzata una denominazione di macro al posto di una semplice label.
03 SYNTAX ERROR
Cifra esadecimale scorretta dopo '$'.
04 MISSING OPERAND

Manca l'argomento dopo .BYT, .DBYT, .WORD o .IF.

05 SYNTAX ERROR

La struttura sintattica di una condizione .IF è sbagliata (per esempio mancano gli spazi vuoti).

06 SYNTAX ERROR

La denominazione manca o non è affidibile.

07 ILLEGAL ADR. MODE

Il modo di indirizzamento desiderato non è permesso.

08 ILLEGAL QUANTITY

Ad un registro ad 8 bit possono essere assegnati solo valori >255.

09 BRANCH OUT OF RANGE

In un salto relativo la differenza di indirizzi è troppo grande.

0B IEC-ERROR

Nell'uso del bus seriale si è verificato un errore (disco pieno, errore di lettura/scrittura, file not found).

0C ILLEGAL OP CODE

È stato trovato un comando illegale o una macro non definita.

0D ILLEGAL FWD REFERENCE

Il valore di un argomento non è noto al momento dell'esecuzione dell'assegnazione del contatore di programma (*=valore), dell'indicatore oggetto (.OBJ valore) o del comando .IF.

0E ILLEGAL QUANTITY

Reset scorretto del contatore di programma in caso di output del codice macchina su disco.

0F SYNTAX ERROR

Definizione macro mancante, scorretta o contenente una lista di parametri scorretta.

10 REDEF'D LABEL

Si è cercato di usare due volte la stessa definizione.

11 ILLEGAL PS OPERATION

Pseudoperazione sconosciuta.

12 SYNTAX ERROR

Numero sbagliato di parametri in una definizione macro.

13 UNBALANCED BLOCKMARKER

Fine di blocco (.^!DEF) senza inizio blocco.

15 OUT OF MEMORY

Ricorsività troppo avanzata in una definizione macro.

16 SYNTAX ERROR

.FILE all'interno di un file caricato in un secondo tempo.

17 SYNTAX ERROR

In .FILE manca il nome del file.

18 OUT OF MEMORY

Troppe chiamate .FILE (mass. 255)

19 OUT OF MEMORY

Condizione .IF troppo nidificata.

1A SYNTAX ERROR

.DEF o .FILE all'interno di una definizione macro.

Ulrich Roller

```
MACRO-ASSEMBLER
PER CHIP SPECIAL
DI ULRICH ROLLER

INDIRIZZO INIZIALE : 31232
INDIRIZZO FINALE   : 40959

7A00:4C,35,7A,4C,78,7A,00,4C,0285
7A08:78,7A,20,7B,89,6C,02,A0,0324
7A10:20,56,7E,20,AD,8D,2C,B9,0333
7A18:9F,10,05,A2,02,4C,98,7E,02BA
7A20:20,26,7A,4C,78,7A,A5,69,030C
7A28:D0,02,C6,6A,C6,69,A5,6A,0440
7A30:48,A5,69,48,60,20,CC,FF,03E9
7A38:AE,94,9F,AC,95,9F,86,57,049E
7A40:84,58,E8,D0,01,C8,8E,96,0481
```

```
7A48:9F,8C,97,9F,A0,00,98,91,042A
7A50:57,8D,C3,9F,8D,C4,9F,A2,04D8
7A58:00,BD,88,9B,F0,06,20,E7,03DD
7A60:85,E8,D0,F5,A9,0A,8D,D1,0543
7A68:9F,A9,80,8D,BC,9F,A9,08,0461
7A70:8D,BD,9F,A9,14,8D,BE,9F,0490
7A78:AD,9A,9F,8D,98,9F,AD,9B,04F2
7A80:9F,8D,99,9F,A9,0D,20,E7,0421
7A88:85,20,80,7B,A9,00,85,6B,0339
7A90:A2,FF,8E,A6,9F,9A,20,7B,04A9
7A98:89,A0,00,20,DD,7E,24,6B,0333
7AA0:10,1B,AD,D1,9F,F0,16,A6,03F4
7AA8:6A,18,65,69,90,01,E8,20,02E9
7AB0:CF,7D,20,CF,FF,A5,D3,85,0537
7AB8:D0,A9,00,85,D3,06,6B,46,0388
7AC0:6B,A2,00,20,CF,FF,C9,0D,03D1
7AC8:F0,06,9D,03,02,E8,D0,F3,0443
7AD0:20,D2,FF,A9,00,9D,03,02,033C
7AD8:AD,DF,7A,AE,E0,7A,2C,02,043C
7AE0:02,85,63,86,64,20,56,7E,02C8
7AE8:F0,B4,90,03,4C,8F,84,AE,0444
7AF0:9A,9F,8E,98,9F,AE,9B,9F,04E6
7AF8:8E,99,9F,AE,03,02,E0,20,0379
7B00:F0,05,06,6B,38,66,6B,18,0287
7B08:20,56,7D,8E,01,02,8C,02,0212
7B10:02,A0,FF,A2,04,C8,B1,63,0423
7B18:29,7F,C9,20,F0,F7,8C,03,0407
7B20:02,88,C8,E8,B1,63,C9,A0,04B7
7B28:D0,02,A9,20,9D,FF,01,C9,0401
7B30:00,D0,EF,8E,00,02,E0,06,0335
7B38:B0,02,A2,00,86,61,85,62,0322
7B40:20,6F,7C,86,5D,84,5E,86,0356
7B48:63,84,64,90,1D,A0,03,C8,0363
7B50:B1,5D,D0,FB,98,38,65,5D,046B
7B58:85,5D,90,02,E6,5E,18,98,0368
7B60:49,FF,65,61,85,61,B0,02,03A6
7B68:C6,62,20,AD,7C,AC,00,02,031F
7B70:C0,06,90,09,88,B9,00,02,02A2
7B78:91,63,98,D0,F7,4C,9E,7A,04B7
7B80:AD,94,9F,AE,95,9F,20,33,0415
7B88:7E,A9,2D,20,E7,85,AD,9A,0427
7B90:9F,AE,9B,9F,20,33,7E,A9,0401
7B98:2F,20,E7,85,38,AD,98,9F,03D7
7BA0:ED,96,9F,48,AD,99,9F,ED,053C
7BA8:97,9F,AA,68,20,CF,7D,A0,0454
7BB0:01,4C,DD,7E,A9,20,20,E7,0378
7BB8:85,A0,02,B1,63,AA,88,B1,041E
7BC0:63,20,CF,7D,2C,BC,9F,30,0386
7BC8:13,A0,03,B1,63,F0,09,A8,036B
7BD0:A9,20,20,E7,85,88,D0,FA,04A7
7BD8:A0,04,D0,68,AE,BD,9F,A0,0486
7BE0:03,B1,63,D0,23,C8,B1,63,03E6
7BE8:F0,64,C9,3B,F0,56,20,E7,04A5
7BF0:85,CA,F0,0B,C8,B1,63,F0,0516
7BF8:55,C9,20,D0,F1,F0,09,C8,04C0
7C00:B1,63,F0,4A,C9,20,D0,F7,04FE
7C08:E8,A9,20,20,E7,85,CA,D0,04D7
7C10:FA,C8,B1,63,F0,38,29,7F,04A6
7C18:C9,20,F0,F5,A2,00,B1,63,0484
7C20:F0,2C,C9,20,D0,0A,C8,B1,0458
7C28:63,C9,3B,F0,0A,88,A9,20,03B2
7C30:20,E7,85,E8,C8,D0,E7,A9,059C
7C38:20,EC,BE,9F,B0,06,20,E7,0426
7C40:85,E8,D0,F5,B1,63,F0,06,053C
7C48:20,E7,85,C8,D0,F6,A9,0D,04D0
7C50:4C,E7,85,18,6D,96,9F,48,03BA
7C58:8A,6D,97,9F,AA,68,B0,0A,03F9
7C60:CD,98,9F,8A,ED,99,9F,B0,0563
7C68:01,60,A2,03,4C,98,7E,AD,0315
7C70:94,9F,AE,95,9F,A0,00,85,043A
7C78:57,86,58,A0,00,B1,57,F0,03CD
7C80:26,A0,01,B1,57,C5,69,C8,03C5
7C88:B1,57,E5,6A,B0,0F,C8,C8,04A6
7C90:B1,57,D0,FB,98,38,65,57,045F
7C98:90,DD,E8,D0,DA,D0,08,88,055F
7CA0:B1,57,C5,69,D0,01,24,18,0343
7CA8:A6,57,A4,58,60,AD,96,9F,043B
7CB0:85,5F,AD,97,9F,85,60,24,03D0
7CB8:62,30,1A,A5,61,A6,62,20,02DA
7CC0:53,7C,18,AD,96,9F,65,61,038F
7CC8:8D,96,9F,AD,97,9F,65,62,046C
7CD0:8D,97,9F,90,20,38,A9,00,0354
7CD8:E5,61,85,61,A9,00,E5,62,041C
7CE0:85,62,38,AD,96,9F,E5,61,0447
7CE8:8D,96,9F,AD,97,9F,E5,62,04EC
7CF0:8D,97,9F,B0,32,18,A5,5F,03C1
7CF8:E5,5D,A8,A5,60,B0,02,C6,0467
7D00:60,E5,5E,AA,E8,90,1F,A5,0489
7D08:5D,85,5F,18,65,61,85,61,0305
7D10:A5,60,65,62,85,62,B1,5F,03C3
7D18:91,61,88,C0,FF,D0,F7,C6,05C6
7D20:60,C6,62,CA,D0,F0,60,38,04AA
7D28:A5,5D,E5,5F,A8,A5,5E,B0,04A1
7D30:02,C6,5E,E5,60,AA,B0,1D,03E2
7D38:A5,5F,85,5D,38,E5,61,85,03E9
7D40:61,A5,5E,E5,62,85,62,B1,0443
7D48:5D,91,61,C8,D0,F9,E6,5E,0524
7D50:E6,62,E8,D0,F2,60,A2,00,04F4
7D58:86,69,86,6A,90,2E,C9,24,038A
7D60:D0,5B,20,7C,7E,90,0C,29,030A
7D68:7F,C9,47,B0,50,C9,41,90,0429
7D70:4C,69,08,29,0F,A2,04,06,01A1
7D78:69,26,6A,B0,47,CA,D0,F7,0481
7D80:65,69,85,69,90,DC,E6,6A,0478
7D88:F0,3A,D0,D6,29,0F,A8,A5,0455
7D90:69,A6,6A,0A,26,6A,B0,2C,02EF
7D98:0A,26,6A,B0,27,65,69,85,02C4
7DA0:69,8A,65,6A,B0,1E,06,69,02FF
7DA8:2A,B0,19,85,6A,98,65,69,0348
7DB0:85,69,90,04,E6,6A,F0,0C,03CE
7DB8:20,7C,7E,90,CF,A6,69,A4,042C
7DC0:6A,4C,82,7E,A2,02,4C,98,033E
7DC8:7E,46,6B,06,6B,90,05,46,027B
7DD0:6B,38,26,6B,85,69,86,6A,0312
```

```
7DD8:A9,00,8D,C6,9F,8D,C7,9F,048E
7DE0:8D,C8,9F,F8,A0,10,06,69,040B
7DE8:26,6A,A2,02,BD,C6,9F,7D,03D3
7DF0:C6,9F,9D,C6,9F,CA,10,F4,0535
7DF8:88,D0,EB,D8,A0,05,A2,04,0466
7E00:0E,C8,9F,2E,C7,9F,2E,C6,03FD
7E08:9F,CA,D0,F4,88,F0,0C,A5,0556
7E10:6B,4A,90,07,AD,C6,9F,29,0387
7E18:F0,F0,E3,A9,03,A2,04,0E,0423
7E20:C8,9F,2E,C7,9F,2E,C6,9F,048E
7E28:2A,CA,D0,F3,20,E7,85,88,04CB
7E30:10,E9,60,48,A9,24,20,E7,0375
7E38:85,68,48,8A,20,40,7E,68,0305
7E40:48,4A,4A,4A,4A,20,49,7E,0257
7E48:68,29,0F,09,30,C9,3A,90,026C
7E50:02,69,06,4C,E7,85,E6,63,0372
7E58:D0,02,E6,64,84,56,A0,00,0396
7E60:B1,63,C9,20,D0,05,C8,10,03AA
7E68:F7,30,0C,98,18,65,63,85,0330
7E70:63,90,11,E6,64,D0,0D,A2,03CD
7E78:05,4C,98,7E,E6,63,D0,02,0382
7E80:E6,64,84,56,A0,00,B1,63,03D8
7E88:A4,56,C9,3A,B0,07,C9,30,03AD
7E90:B0,02,38,24,18,29,FF,60,02AE
7E98:8A,A2,FF,9A,48,20,7B,89,0431
7EA0:68,F0,0D,A8,A2,FF,E8,BD,0553
7EA8:27,9C,10,FA,88,D0,F7,E8,0504
7EB0:BD,27,9C,29,7F,20,E7,85,03B4
7EB8:BD,27,9C,10,F2,A0,02,20,0344
7EC0:DD,7E,2C,A6,9F,30,0E,A0,03AA
7EC8:03,20,DD,7E,AD,A2,9F,AE,041A
7ED0:A3,9F,20,CF,7D,A9,0D,20,0384
7ED8:E7,85,4C,90,7A,48,8A,48,03DC
7EE0:A2,FF,98,F0,09,E8,BD,C1,0598
7EE8:9B,D0,FA,88,D0,F7,E8,BD,0659
7EF0:C1,9B,F0,05,20,E7,85,D0,04AD
7EF8:F5,68,AA,68,60,20,56,7E,03C3
7F00:20,1F,84,86,63,84,64,A5,0339
7F08:63,C5,59,A5,64,E5,5A,B0,0479
7F10:12,20,A5,88,20,B4,7B,98,0346
7F18:38,65,63,85,63,90,E8,E6,0446
7F20:64,B0,E4,4C,90,7A,20,56,03C4
7F28:7E,A2,00,B0,0E,20,56,7D,02D1
7F30:A6,69,A4,6A,F0,05,A2,02,03B6
7F38:4C,98,7E,8E,D1,9F,4C,90,043C
7F40:7A,20,56,7E,20,1F,84,38,0269
7F48:A5,57,E5,59,85,61,A5,58,041D
7F50:E5,5A,85,62,A5,59,85,5D,0406
7F58:A5,5A,85,5E,20,AD,7C,4C,0377
7F60:90,7A,20,56,7E,20,1F,84,02C1
7F68:20,82,7E,C9,2C,D0,44,20,0349
7F70:56,7E,B0,3F,20,56,7D,A5,035B
7F78:6A,D0,73,85,55,A5,57,48,03CB
7F80:A5,58,48,A5,5B,48,A5,5C,038E
7F88:48,20,B8,7F,A0,00,B1,57,0347
7F90:F0,0D,A0,01,A5,5B,D1,57,03C6
7F98:C8,A5,5C,F1,57,B0,4F,68,0478
7FA0:85,5C,68,85,5B,68,85,58,036E
7FA8:68,85,57,E6,55,20,B8,7F,03D6
7FB0:4C,90,7A,A2,01,4C,98,7E,035B
7FB8:A5,57,C5,59,A5,58,E5,5A,0456
7FC0:B0,2B,A5,55,F0,0B,A0,01,0371
7FC8:A5,5B,91,57,A5,5C,C8,91,0442
7FD0:57,18,A5,5B,65,69,85,5B,031D
7FD8:90,04,E6,5C,F0,10,A0,00,0376
7FE0:B1,57,18,65,57,85,57,90,0348
7FE8:CF,E6,58,D0,CB,60,A2,02,04AC
7FF0:4C,98,7E,0E,BC,9F,38,6E,0371
7FF8:BC,9F,20,56,7E,F0,22,B0,0411
8000:23,20,56,7D,98,D0,E7,8E,03F3
8008:BD,9F,20,5C,7E,F0,12,C9,0421
8010:2C,D0,A0,20,56,7E,B0,9B,03DB
8018:20,56,7D,98,D0,D0,8E,BE,0477
8020:9F,4C,90,7A,C9,43,D0,8B,045C
8028:0E,BC,9F,4E,BC,9F,90,F1,0493
8030:20,8E,80,8C,02,02,20,56,0234
8038:7E,20,1F,84,A5,57,85,63,0325
8040:A5,58,85,64,A9,00,85,57,036B
8048:85,58,85,5B,46,5C,A5,63,0367
8050:C5,59,A5,64,E5,5A,B0,1A,0430
8058:20,BF,80,90,EF,E6,57,D0,04EB
8060:02,E6,58,24,5C,30,F1,66,0347
8068:5C,20,A5,88,20,B4,7B,4C,0344
8070:58,80,A9,2A,20,E7,85,20,0357
8078:E7,85,A9,20,20,E7,85,A5,0466
8080:57,A6,58,20,CF,7D,A9,0D,0377
8088:20,E7,85,4C,90,7A,A2,00,0384
8090:20,56,7E,85,5B,20,7C,7E,02EE
8098:F0,20,C9,A0,D0,02,A9,20,0414
80A0:C5,5B,F0,06,9D,03,02,E8,03A0
80A8:D0,EB,A9,00,9D,03,02,A0,03A6
80B0:FF,C8,B9,03,02,C9,20,F0,045E
80B8:F8,60,A2,04,4C,98,7E,A5,0405
80C0:5B,D0,2A,A0,03,84,5B,AE,0385
80C8:02,02,F0,21,B1,63,CD,02,02F8
80D0:02,90,1A,C8,BD,03,02,F0,0326
80D8:2C,C9,5C,D0,08,B1,63,F0,042D
80E0:0C,C9,20,F0,08,D1,63,D0,03F1
80E8:04,E8,C8,D0,E7,E6,5B,A2,054E
80F0:00,A4,5B,B1,63,D0,DD,85,0445
80F8:5B,98,38,65,63,85,63,90,036B
8100:02,E6,64,18,24,38,60,A2,02C2
8108:02,2C,A2,01,4C,98,7E,20,0253
8110:41,82,90,F3,38,A5,59,E5,0461
8118:5B,85,61,A5,5A,E5,5C,85,0406
8120:62,A5,57,85,5D,C5,5B,A5,0405
8128:58,85,5E,E5,5C,B0,19,A5,03EA
8130:5B,65,61,85,5B,A5,5C,65,0367
8138:62,85,5C,18,A5,59,65,61,031F
8140:85,59,A5,5A,65,62,85,5A,0383
8148:20,AD,7C,A5,5B,C5,59,A5,040C
8150:5C,E5,5A,B0,32,A0,00,B1,03CE
8158:5B,91,57,C8,A5,69,91,57,0401
8160:C8,A5,6A,91,57,C8,B1,5B,0493
```

```
8168:91,57,C8,B1,5B,91,57,D0,0474
8170:F9,98,38,65,5B,85,5B,90,03F9
8178:02,E6,5C,38,98,65,57,85,0355
8180:57,90,C8,E6,58,B0,C4,4C,04AD
8188:90,7A,A2,02,4C,98,7E,20,0330
8190:41,82,B0,1D,A5,5B,85,57,036C
8198:A5,5C,85,58,A5,69,85,5B,03CC
81A0:A5,6A,85,5C,A2,00,86,69,0381
81A8:E8,86,55,20,B8,7F,4C,90,03F6
81B0:7A,A5,5B,C5,59,A5,5C,E5,047E
81B8:5A,B0,CC,A0,00,B1,5B,85,0407
81C0:61,84,62,A5,57,85,5D,A5,03CA
81C8:58,85,5E,A5,5B,C5,57,A5,03FC
81D0:5C,E5,58,90,0B,18,A5,5B,034C
81D8:65,61,85,5B,90,02,E6,5C,037A
81E0:20,AD,7C,A0,00,B1,5B,91,0386
81E8:57,C8,A5,69,91,57,C8,A5,0482
81F0:6A,91,57,C8,B1,5B,91,57,040E
81F8:C8,B1,5B,91,57,D0,F9,A8,052D
8200:38,F1,5B,85,61,88,84,62,03D8
8208:18,A0,00,B1,5B,65,5B,85,0309
8210:5D,98,65,5C,85,5E,A5,5B,0399
8218:C5,57,A5,5C,E5,58,90,0D,03F7
8220:18,B1,57,65,57,85,57,90,0348
8228:0F,E6,58,B0,0B,A5,59,38,033E
8230:F1,5B,85,59,B0,02,C6,5A,03FC
8238:20,AD,7C,4C,B1,81,4C,90,03A3
8240:7A,20,56,7E,20,1F,84,86,02B7
8248:5B,84,5C,20,5C,7E,C9,2C,032A
8250:D0,3D,20,56,7E,20,56,7D,02F4
8258:20,6F,7C,A0,00,B1,57,F0,03A3
8260:1B,C8,A5,69,D1,57,C8,A5,0486
8268:6A,F1,57,90,0F,A0,00,B1,03A2
8270:57,18,65,57,85,57,90,E3,037A
8278:E6,58,B0,DF,A5,5B,C5,57,04E9
8280:A5,5C,E5,58,B0,08,A5,57,03F2
8288:C5,59,A5,58,E5,5A,60,A2,045C
8290:01,2C,A2,04,2C,A2,02,4C,01EF
8298:98,7E,20,8E,80,AD,03,02,02F6
82A0:F0,F3,8E,01,02,8C,02,02,0304
82A8:A0,FF,E8,C8,20,7C,7E,F0,0559
82B0:E1,C9,A0,D0,02,A9,20,9D,0482
82B8:03,02,C5,5B,D0,EC,A9,00,038A
82C0:9D,03,02,AE,01,02,98,ED,02D8
82C8:01,02,8D,01,02,A0,FF,E8,031A
82D0:C8,BD,03,02,C9,20,F0,F7,045A
82D8:38,98,ED,02,02,8D,00,02,0250
82E0:46,5D,20,56,7E,90,0F,F0,0326
82E8:0D,C9,2A,D0,A2,66,5D,20,0355
82F0:56,7E,F0,02,B0,99,20,1F,034E
82F8:84,86,63,84,64,A6,69,A4,0408
8300:6A,86,59,84,5A,A5,5D,85,03AE
8308:5C,A9,00,85,57,85,58,A0,035E
8310:00,84,5B,B1,63,F0,0C,C8,03B7
8318:A5,59,D1,63,C8,A5,5A,F1,04EA
8320:63,B0,03,4C,72,80,20,BF,0333
8328:80,90,E4,84,5F,24,5C,10,0367
8330:13,A9,2A,20,E7,85,20,B4,0346
8338:7B,20,BD,88,C9,59,F0,04,03F6
8340:C9,0D,D0,E2,E6,57,D0,02,0497
8348:E6,58,18,A5,5F,65,63,85,03A7
8350:5D,A9,00,65,64,85,5E,A9,035B
8358:00,85,62,A0,03,C4,5B,90,0339
8360:0E,18,B1,63,6D,00,02,91,023A
8368:63,38,AD,00,02,49,FF,6D,02FF
8370:01,02,85,61,10,02,C6,62,0223
8378:A0,00,18,71,63,30,63,20,023F
8380:AD,7C,A2,FF,E8,BD,03,02,0474
8388:D0,FA,E8,A4,5B,C0,04,B0,0525
8390:0A,CA,E8,BD,03,02,C9,20,0367
8398:F0,F8,C8,BD,03,02,F0,06,0468
83A0:91,63,C8,E8,D0,F5,88,84,0575
83A8:5B,A0,03,C8,B1,63,F0,48,0412
83B0:C9,20,F0,F7,C0,04,F0,2A,04AE
83B8:98,18,65,63,85,5D,A9,00,0303
83C0:65,64,85,5E,98,E9,04,49,037A
83C8:FF,85,61,A2,FF,86,62,49,04B7
83D0:FF,38,A0,03,71,63,91,63,03A2
83D8:18,A5,5B,65,61,85,5B,20,02DE
83E0:AD,7C,A0,03,C8,B1,63,D0,0478
83E8:FB,C8,98,A0,00,91,63,20,040F
83F0:B4,7B,20,A5,88,4C,26,83,0371
83F8:A0,03,C8,B1,63,D0,FB,98,04E2
8400:49,FF,85,61,A9,FF,85,62,04BD
8408:38,98,65,63,85,5D,A9,00,0323
8410:65,64,85,5E,20,AD,7C,A0,0395
8418:05,20,DD,7E,4C,0F,83,A2,0300
8420:00,86,69,86,6A,20,5C,7E,02D9
8428:90,15,C9,2D,F0,11,86,5B,037D
8430:86,5C,20,6F,7C,86,59,84,0350
8438:5A,C6,69,C6,6A,D0,25,20,03CE
8440:56,7D,86,5B,84,5C,20,6F,0323
8448:7C,86,59,84,5A,20,5C,7E,0333
8450:90,0F,C9,2D,D0,0E,A9,FF,041B
8458:85,69,85,6A,20,56,7E,B0,0381
8460:03,20,56,7D,A0,00,B1,59,02A0
8468:F0,1B,C8,A5,69,D1,59,C8,04D3
8470:A5,6A,F1,59,90,0F,A0,00,0398
8478:B1,59,18,65,59,85,59,90,034E
8480:E3,E6,5A,B0,DF,A6,57,A4,0553
8488:58,60,A2,01,4C,98,7E,20,02DD
8490:F1,84,90,05,A2,01,4C,98,0391
8498:7E,0A,65,55,AA,BD,A8,84,03D5
84A0:48,BD,A7,84,48,60,2C,FC,0400
84A8:7E,2C,25,7F,2C,40,7F,2C,0265
84B0:61,7F,2C,F2,7F,2C,2F,80,0358
84B8:2C,99,82,2C,0E,81,2C,8E,02BC
84C0:81,2C,BD,86,2C,29,85,2C,02F6
84C8:05,7A,2C,09,7A,2C,2C,87,020D
84D0:2C,E8,87,2C,42,88,2C,12,02CF
84D8:88,2C,27,88,2C,34,7A,2C,0269
84E0:86,86,2C,93,8A,2C,F3,8A,03FE
84E8:2C,0F,7A,2C,F9,8A,2C,C7,0357
84F0:87,A2,00,86,55,CA,A0,FF,046D
```

```
84F8:E8,C8,BD,B8,9C,F0,19,51,051B
8500:63,D0,07,BD,B8,9C,10,F0,044B
8508:30,12,29,7F,F0,0E,E6,55,0323
8510:BD,B8,9C,30,E1,E8,D0,F8,05D2
8518:38,A5,55,60,98,18,65,63,030A
8520:85,63,90,02,E6,64,18,A5,0381
8528:55,60,20,39,8A,20,56,7E,028C
8530:D0,03,4C,D3,85,90,04,C9,03D4
8538:24,D0,1F,20,56,7D,86,57,02E3
8540:84,58,AE,9A,9F,AC,9B,9F,04A9
8548:20,5C,7E,F0,13,C9,2D,D0,03C3
8550:09,20,56,7E,90,07,C9,24,0281
8558:F0,03,4C,DC,85,20,56,7D,0393
8560:86,59,84,5A,8A,38,E5,57,03BB
8568:85,5B,98,E5,58,85,5C,90,0426
8570:6E,05,5B,F0,6A,AD,94,9F,0408
8578:85,5D,AD,95,9F,85,5E,AD,0453
8580:96,9F,85,5F,E5,5D,AA,AD,04B2
8588:97,9F,85,60,E5,5E,90,52,0440
8590:E4,5B,E5,5C,B0,4C,38,A5,0459
8598:57,8D,94,9F,E5,5D,85,61,043F
85A0:A5,58,8D,95,9F,E5,5E,85,0486
85A8:62,A5,59,8D,9A,9F,8D,98,044B
85B0:9F,A5,5A,8D,9B,9F,8D,99,048B
85B8:9F,24,62,30,06,20,F5,7C,02EC
85C0:4C,D3,85,A9,00,38,E5,61,03CB
85C8:85,61,A9,00,E5,62,85,62,03BD
85D0:20,27,7D,20,39,8A,20,80,0247
85D8:7B,4C,90,7A,A2,01,2C,A2,0342
85E0:02,2C,A2,03,4C,98,7E,48,027D
85E8:AD,C2,9F,D0,04,68,4C,D2,0468
85F0:FF,8D,C0,9F,AD,C3,9F,D0,05CA
85F8:04,68,4C,92,89,AD,C4,9F,03E3
8600:D0,56,A9,3F,8D,C4,9F,A9,04A7
8608:51,8D,C5,9F,98,48,A9,20,03EB
8610:A0,20,20,7A,86,88,D0,FA,0432
8618:A0,04,20,DD,7E,8A,48,AD,039E
8620:C3,9F,48,A0,08,A9,00,85,0380
8628:56,AA,F8,0E,C3,9F,65,56,0423
8630:85,56,90,01,E8,88,D0,F3,049F
8638:D8,20,3A,7E,68,8D,C3,9F,0407
8640:EE,C3,9F,68,AA,68,A8,A9,051B
8648:2D,20,7A,86,20,7A,86,20,028D
8650:7A,86,20,5D,86,20,5D,86,0306
8658:68,C9,0D,D0,1D,CE,C4,9F,045C
8660:D0,0E,8A,48,A9,0D,A2,09,0311
8668:20,92,89,CA,D0,FA,68,AA,04E1
8670:A9,51,8D,C5,9F,A9,0D,4C,03ED
8678:92,89,48,CE,C5,9F,D0,03,0468
8680:20,5D,86,68,4C,92,89,20,02F2
8688:A5,86,20,56,7E,F0,07,B0,03C6
8690:0B,20,56,7D,A5,69,8D,C3,035C
8698:9F,4C,90,7A,C9,2B,F0,F9,04D2
86A0:A2,01,4C,98,7E,AE,C2,9F,0414
86A8:F0,0E,8E,C0,9F,AE,C4,9F,04FC
86B0:F0,06,20,5D,86,4C,AD,86,0378
86B8:A9,00,8D,C4,9F,60,20,CC,03E5
86C0:88,86,57,A9,24,8D,03,02,02C4
86C8:A0,00,8C,04,02,8A,A2,01,025F
86D0:8E,BF,9F,20,08,89,20,0D,02CA
86D8:87,20,0D,87,20,0D,87,20,020F
86E0:0D,87,20,0D,87,48,20,0D,01BD
86E8:87,AA,68,20,CF,7D,A9,20,03CE
86F0:20,E7,85,20,0D,87,F0,05,0335
86F8:20,E7,85,D0,F6,A9,0D,20,0428
8700:E7,85,20,A5,88,4C,DC,86,0467
8708:A2,06,4C,98,7E,20,C4,89,0377
8710:48,A5,90,F0,16,68,68,68,03BB
8718:A9,0D,20,E7,85,20,7B,89,0366
8720:A6,57,A9,BF,24,90,D0,62,044B
8728:4C,90,7A,68,60,20,CC,88,0392
8730:A0,0F,20,5C,7E,F0,53,C9,03B5
8738:2C,D0,12,8A,48,20,56,7E,02D4
8740:B0,7B,20,56,7D,A5,6A,D0,03FD
8748:71,A4,69,68,AA,86,57,84,03F1
8750:58,20,ED,88,20,44,89,A9,0383
8758:00,85,90,A5,57,20,B1,FF,03E1
8760:A5,58,09,60,20,93,FF,24,033C
8768:90,30,55,A2,00,BD,03,02,0279
8770:F0,06,20,A8,FF,E8,D0,F5,056A
8778:20,7C,7E,C9,3B,F0,05,A9,03BC
8780:0D,20,A8,FF,20,AE,FF,4C,03ED
8788:90,7A,20,44,89,A9,00,85,0325
8790:90,8A,20,B1,FF,A9,6F,20,0422
8798:93,FF,24,90,30,22,20,AE,0366
87A0:FF,8A,20,B4,FF,A9,6F,20,0494
87A8:96,FF,20,A5,FF,20,D2,FF,054A
87B0:A5,90,F0,F6,20,AB,FF,4C,0531
87B8:90,7A,A2,02,2C,A2,01,2C,02A9
87C0:A2,06,2C,A2,07,4C,98,7E,02DF
87C8:AD,C2,9F,D0,F0,A9,00,8D,0504
87D0:03,02,A9,04,A0,00,A2,03,01F7
87D8:8E,C2,9F,20,08,89,20,56,0316
87E0:7E,4C,8F,84,A2,0A,4C,98,036D
87E8:7E,20,CC,88,86,61,20,ED,03E6
87F0:88,F0,F1,AD,94,9F,AE,95,058C
87F8:9F,85,5D,86,5E,AD,9A,9F,044B
8800:AE,9B,9F,8D,98,9F,8E,99,04D3
8808:9F,85,5F,86,60,20,F1,89,0403
8810:4C,90,7A,20,CC,88,86,61,03B1
8818:20,ED,88,F0,C7,20,39,8A,042F
8820:18,A5,5F,A6,60,4C,F9,87,03EE
8828:AD,94,9F,AE,95,9F,85,63,04AA
8830:86,64,A9,01,A0,00,91,63,0328
8838:20,39,8A,4C,90,7A,A2,0A,02E5
8840:4C,98,7E,20,CC,88,86,5D,03B9
8848:20,ED,88,F0,F1,20,56,7E,046A
8850:20,1F,84,A5,5D,A0,01,A2,0308
8858:01,8E,C0,9F,20,08,89,38,02D7
8860:A5,59,E5,57,AA,A5,5A,E5,04C8
8868:58,48,8A,20,92,89,68,20,02ED
8870:92,89,A5,57,C5,59,A5,58,0432
8878:E5,5A,B0,13,A0,00,B1,57,03AA
8880:20,92,89,A5,90,D0,19,E6,043F
```

```
8888:57,D0,E7,E6,58,D0,E3,A9,05A8
8890:00,20,92,89,A2,01,20,57,0255
8898:89,A5,90,D0,03,4C,90,7A,03E7
88A0:A6,5D,4C,8A,87,98,48,8A,03CA
88A8:48,78,20,9F,FF,58,68,AA,03E8
88B0:68,A8,20,E4,FF,C9,03,F0,04CF
88B8:0E,C9,5F,D0,09,20,E4,FF,0412
88C0:F0,FB,C9,03,F0,01,60,A2,04AA
88C8:00,4C,98,7E,A2,08,20,56,0282
88D0:7E,B0,09,20,56,7D,A5,6A,0339
88D8:D0,0B,A6,69,E0,10,B0,05,038F
88E0:E0,04,90,01,60,A2,07,2C,02AA
88E8:A2,01,4C,98,7E,20,5C,7E,02FF
88F0:C9,22,D0,F4,A2,FF,E8,20,0558
88F8:7C,7E,F0,EC,9D,03,02,49,03C1
8900:22,D0,F3,9D,03,02,8A,60,0371
8908:9D,C8,9F,98,09,60,AC,03,03B4
8910:02,F0,02,09,F0,9D,CC,9F,03F5
8918:29,FF,10,27,20,44,89,A9,02F5
8920:00,85,90,BD,C8,9F,20,B1,040A
8928:FF,BD,CC,9F,20,93,FF,24,04FD
8930:90,30,5A,A2,00,BD,03,02,027E
8938:F0,06,20,A8,FF,E8,D0,F5,056A
8940:20,AE,FF,60,AD,C1,9F,F0,052A
8948:31,0A,A9,00,8D,C1,9F,90,0361
8950:03,4C,AB,FF,4C,AE,FF,20,0412
8958:44,89,BD,CC,9F,10,13,BD,03D5
8960:C8,9F,20,B1,FF,BD,CC,9F,055F
8968:29,0F,09,E0,20,93,FF,20,02F3
8970:AE,FF,A9,00,9D,C8,9F,9D,04F7
8978:CC,9F,60,A2,04,20,57,89,0371
8980:CA,D0,FA,8E,BF,9F,8E,C0,05CE
8988:9F,8E,C2,9F,60,A2,06,4C,03E2
8990:98,7E,48,AD,C0,9F,CD,C1,04F8
8998:9F,F0,25,8A,48,20,44,89,0373
89A0:AE,C0,9F,8E,C1,9F,A9,00,04A4
89A8:85,90,BD,C8,9F,20,B1,FF,0509
89B0:BD,CC,9F,29,0F,09,60,20,02E9
89B8:93,FF,24,90,30,CF,68,AA,0457
89C0:68,4C,A8,FF,AD,BF,9F,09,046F
89C8:80,CD,C1,9F,F0,20,8A,48,048F
89D0:20,44,89,AD,BF,9F,AA,09,03AB
89D8:80,8D,C1,9F,BD,C8,9F,20,04B1
89E0:B4,FF,BD,CC,9F,29,0F,09,041C
89E8:60,20,96,FF,68,AA,4C,A5,0418
89F0:FF,A5,61,A0,00,A2,01,8E,03D6
89F8:BF,9F,20,08,89,20,C4,89,037C
8A00:A5,90,4A,4A,B0,26,20,C4,0383
8A08:89,A5,5D,C5,5F,A5,5E,E5,0497
8A10:60,B0,1E,20,C4,89,A0,00,033B
8A18:91,5D,E6,5D,D0,02,E6,5E,0447
8A20:24,90,50,E5,A2,01,20,57,0303
8A28:89,4C,39,8A,A2,08,4C,98,0326
8A30:7E,20,39,8A,A2,03,4C,98,02EA
8A38:7E,AD,94,9F,AE,95,9F,85,04C5
8A40:5F,86,60,85,5D,86,5E,18,0323
8A48:ED,98,9F,8A,ED,99,9F,B0,0583
8A50:26,A5,5D,85,5F,86,60,A0,0392
8A58:00,B1,5F,F0,21,A0,04,B1,0376
8A60:5F,F0,05,C8,10,F9,30,0F,0364
8A68:C8,30,0C,98,A0,00,91,5D,032A
8A70:65,5D,90,CF,E8,B0,CC,A9,052E
8A78:00,A0,00,91,5F,38,A6,5F,02CD
8A80:A4,60,E8,D0,01,C8,8E,96,04A9
8A88:9F,8C,97,9F,B0,01,60,A2,0414
8A90:05,4C,98,7E,A2,00,86,69,02F8
8A98:86,6A,20,56,7E,B0,03,20,02B7
8AA0:56,7D,20,6F,7C,8E,A0,9F,03AB
8AA8:8C,A1,9F,A0,00,B1,57,F0,0464
8AB0:40,20,5C,7E,F0,0A,29,7F,02DC
8AB8:C9,4C,F0,07,C9,4D,F0,06,0418
8AC0:A9,00,2C,A9,01,2C,A9,11,0265
8AC8:8D,BA,9F,AD,9A,9F,AE,9B,0515
8AD0:9F,8D,98,9F,8E,99,9F,AD,04D6
8AD8:96,9F,AE,97,9F,8D,9C,9F,04E1
8AE0:8E,9D,9F,A9,00,8D,A6,9F,0445
8AE8:20,00,8B,EE,A6,9F,20,00,02FE
8AF0:8B,4C,90,7A,20,28,91,4C,0306
8AF8:90,7A,20,05,9B,4C,90,7A,0320
8B00:AD,A0,9F,85,63,AD,A1,9F,04C1
8B08:85,64,A9,00,8D,9E,9F,8D,03E9
8B10:9F,9F,8D,A7,9F,8D,A8,9F,04E5
8B18:8D,A9,9F,8D,AA,9F,8D,BB,04F3
8B20:9F,8D,AB,9F,85,65,8D,AC,0499
8B28:9F,85,66,8D,B6,9F,8D,B7,04B0
8B30:9F,20,55,96,20,A5,88,20,0317
8B38:82,7E,08,20,7C,7E,28,F0,033A
8B40:03,4C,F0,8B,20,6D,8F,90,0376
8B48:03,4C,05,8C,20,03,90,AD,0240
8B50:B8,9F,85,61,10,03,20,9C,030C
8B58:90,AC,A6,9F,F0,08,A0,00,0419
8B60:B1,5D,09,20,91,5D,20,76,02BB
8B68:96,20,5C,7E,C9,5F,F0,18,03C0
8B70:C9,3D,D0,57,24,61,30,10,02F2
8B78:AE,A6,9F,D0,0B,A2,0A,2C,03A6
8B80:A2,02,20,88,8F,4C,B5,8B,0367
8B88:A9,20,24,61,D0,F2,A5,5D,0412
8B90:85,61,A5,5E,85,62,20,56,0346
8B98:7E,20,AD,8D,A0,00,B1,61,038A
8BA0:29,EF,2C,B9,9F,10,02,09,02B7
8BA8:10,91,61,C8,A5,69,91,61,03CA
8BB0:C8,A5,6A,91,61,A9,00,8D,03FF
8BB8:AD,9F,20,F1,8C,20,A8,8C,043D
8BC0:20,EA,8F,B0,03,4C,31,8B,0354
8BC8:4C,34,9A,A9,20,24,61,30,0298
8BD0:12,D0,08,AE,A6,9F,D0,0B,03B8
8BD8:A2,0A,2C,A2,02,20,88,8F,02B3
8BE0:4C,F0,8B,A0,01,AD,AB,9F,045F
8BE8:91,5D,C8,AD,AC,9F,91,5D,049C
8BF0:20,5C,7E,F0,C0,20,6D,8F,03C6
8BF8:B0,0B,20,12,93,B0,03,4C,027F
8C00:6D,93,4C,D5,91,C9,3B,F0,04A6
8C08:AC,C9,2E,D0,03,4C,F0,94,0446
8C10:C9,2A,F0,08,A2,0C,20,88,0341
```

```
8C18:8F,4C,B5,8B,20,56,7E,C9,03D8
8C20:3D,D0,F1,20,56,7E,20,AD,03BF
8C28:8D,A9,88,2C,B9,9F,F0,0A,043C
8C30:A2,0D,20,88,8F,A2,00,4C,02D4
8C38:98,7E,38,A5,69,ED,AB,9F,0493
8C40:85,5D,A5,6A,ED,AC,9F,85,04AE
8C48:5E,05,5D,F0,33,2C,BA,9F,0368
8C50:10,2E,A2,0E,90,DC,A9,02,0305
8C58:8D,C0,9F,A2,00,A0,00,A9,03D7
8C60:00,20,92,89,E8,D0,01,C8,03BC
8C68:E4,5D,98,E5,5E,90,F0,A5,0541
8C70:90,F0,0D,AD,BA,9F,29,7F,043B
8C78:8D,BA,9F,A2,0B,20,88,8F,03CA
8C80:18,A5,69,8D,AB,9F,A5,6A,040C
8C88:8D,AC,9F,A5,65,65,5D,85,0429
8C90:65,A5,66,65,5E,85,66,20,033E
8C98:5C,7E,F0,09,C9,3B,F0,05,03CC
8CA0:A2,06,20,88,8F,4C,B5,8B,036B
8CA8:AD,A6,9F,F0,32,AD,BA,9F,051A
8CB0:4A,90,2C,AD,BB,9F,D0,28,0405
8CB8:AD,A2,9F,AE,A3,9F,20,C9,04C7
8CC0:7D,A9,20,20,E7,85,A5,64,03DB
8CC8:48,A5,63,48,AD,A4,9F,85,040D
8CD0:63,AD,A5,9F,85,64,20,C4,0421
8CD8:7B,68,85,63,68,85,64,60,037C
8CE0:AE,AD,9F,F0,FA,A9,10,2C,04C9
8CE8:BA,9F,F0,F3,A9,0D,4C,E7,0525
8CF0:85,AD,A6,9F,D0,03,4C,9D,0433
8CF8:8D,2C,BA,9F,50,1A,AC,AD,03D5
8D00:9F,F0,15,88,B9,AE,9F,91,04C3
8D08:65,88,10,F8,18,A5,65,6D,0384
8D10:AD,9F,85,65,90,02,E6,66,0414
8D18:2C,BA,9F,10,27,AC,AD,9F,03B4
8D20:F0,22,A9,02,8D,C0,9F,A2,044B
8D28:00,BD,AE,9F,20,92,89,E8,042D
8D30:88,D0,F6,A5,90,F0,0D,A2,0522
8D38:0B,20,88,8F,AD,BA,9F,29,0371
8D40:7F,8D,BA,9F,AD,BA,9F,4A,04B5
8D48:90,53,AE,BB,9F,F0,0C,A9,0490
8D50:10,2C,BA,9F,F0,47,AE,AD,0427
8D58:9F,F0,51,AE,AD,9F,F0,33,04FD
8D60:20,A5,88,AD,AB,9F,AE,AC,049E
8D68:9F,20,3A,7E,A9,2D,20,E7,0354
8D70:85,A2,00,A9,20,20,E7,85,037C
8D78:BD,AE,9F,20,40,7E,E8,EC,04BC
8D80:AD,9F,90,EF,A9,03,ED,AD,0511
8D88:9F,85,55,90,1F,0A,65,55,02EC
8D90:AA,E8,2C,A2,0F,A9,20,20,0358
8D98:E7,85,CA,10,FA,18,AD,AB,04B0
8DA0:9F,6D,AD,9F,8D,AB,9F,90,04BF
8DA8:03,EE,AC,9F,60,A9,00,8D,03D2
8DB0:B9,9F,20,5C,7E,C9,3C,F0,0447
8DB8:07,C9,3E,D0,0F,2E,B9,9F,0373
8DC0:AD,B9,9F,2A,09,40,8D,B9,03BE
8DC8:9F,20,56,7E,20,3C,8E,20,029D
8DD0:82,7E,C9,2B,F0,04,C9,2D,03DE
8DD8:D0,2F,A5,6A,48,A5,69,48,03AC
8DE0:20,82,7E,48,20,7C,7E,20,02A2
8DE8:3C,8E,68,C9,2D,F0,0D,68,038D
8DF0:65,69,85,69,68,65,6A,85,0378
8DF8:6A,4C,CF,8D,68,E5,69,85,044D
8E00:69,68,E5,6A,85,6A,4C,CF,042A
8E08:8D,A2,00,AD,B9,9F,0A,0A,0348
8E10:0A,0A,29,80,0D,B9,9F,8D,02AF
8E18:B9,9F,4A,B0,07,4A,90,08,033B
8E20:A5,6A,85,69,86,6A,B0,0B,03A8
8E28:A9,90,2C,B9,9F,D0,0C,A5,043E
8E30:6A,D0,08,A9,40,0D,B9,9F,0390
8E38:8D,B9,9F,60,A9,00,85,69,03DC
8E40:85,6A,20,82,7E,B0,30,A8,0397
8E48:A5,6A,AA,A5,69,0A,26,6A,0361
8E50:0A,26,6A,18,65,69,85,69,026E
8E58:8A,65,6A,06,69,2A,AA,18,02B4
8E60:98,29,0F,65,69,85,69,8A,0316
8E68:69,00,85,6A,20,7C,7E,90,0302
8E70:D6,A6,69,A4,6A,18,60,C9,0434
8E78:24,D0,31,20,94,8E,B0,4A,0361
8E80:A0,04,06,69,26,6A,88,D0,02FB
8E88:F9,05,69,85,69,20,94,8E,0397
8E90:90,EE,B0,DD,20,7C,7E,90,04B5
8E98:0C,29,7F,C9,47,B0,0A,C9,0347
8EA0:41,90,06,69,08,29,0F,18,0198
8EA8:60,4C,82,7E,C9,25,D0,28,0392
8EB0:20,7C,7E,4A,49,18,D0,12,02A7
8EB8:26,69,26,6A,20,7C,7E,4A,0283
8EC0:49,18,F0,F4,A6,69,A4,6A,0462
8EC8:18,60,A2,03,20,88,8F,0E,0262
8ED0:B9,9F,38,6E,B9,9F,38,60,03EE
8ED8:C9,2A,D0,0F,AE,AB,9F,AC,0476
8EE0:AC,9F,86,69,84,6A,20,7C,03C4
8EE8:7E,18,60,20,6D,8F,B0,36,02F8
8EF0:20,03,90,86,69,84,6A,20,02B0
8EF8:41,8F,98,18,65,63,85,63,0330
8F00:90,02,E6,64,A9,20,2C,B8,0389
8F08:9F,D0,21,10,05,AD,A6,9F,0397
8F10:D0,17,AD,B8,9F,29,98,AE,045A
8F18:A6,9F,F0,02,49,10,0D,B9,0356
8F20:9F,8D,B9,9F,18,60,A2,04,03A2
8F28:2C,A2,01,2C,A2,02,20,88,0247
8F30:8F,AD,B9,9F,09,80,8D,B9,0463
8F38:9F,A9,00,85,69,85,6A,38,035D
8F40:60,A0,00,B1,63,20,6D,8F,0330
8F48:B0,22,C8,B1,63,20,7F,8F,03DC
8F50:90,F8,20,6D,8F,90,F3,C9,04F0
8F58:2F,F0,EF,C9,2E,F0,EB,C9,05A9
8F60:24,F0,E7,C9,3F,F0,E3,C9,059F
8F68:23,F0,DF,18,60,48,C9,5C,03D7
8F70:18,F0,0A,29,7F,C9,41,90,0354
8F78:03,C9,5B,24,38,68,60,C9,0314
8F80:30,90,03,C9,3A,24,38,60,0282
8F88:8A,20,40,7E,BD,84,9E,AA,03F1
8F90:A0,FF,C8,B9,A5,9E,D0,FA,062D
8F98:CA,D0,F7,A9,20,20,E7,85,04E6
8FA0:C8,B9,A5,9E,D0,F7,2C,A6,055D
```

```
8FA8:9F,30,31,A2,00,BD,A5,9E,03A2
8FB0:20,E7,85,E8,BD,A5,9E,D0,0544
8FB8:F7,A5,6A,48,A5,69,48,AD,0451
8FC0:A2,9F,AE,A3,9F,20,CF,7D,049D
8FC8:68,85,69,68,85,6A,A9,2F,0385
8FD0:20,E7,85,AD,AB,9F,AE,AC,04DD
8FD8:9F,20,3A,7E,A9,0D,20,E7,0334
8FE0:85,EE,B6,9F,D0,03,EE,B7,0540
8FE8:9F,60,A0,00,B1,63,F0,03,03A6
8FF0:C8,D0,F9,98,38,65,63,85,04AE
8FF8:63,90,02,E6,64,A0,00,98,0377
9000:D1,63,60,20,41,8F,88,C0,03CC
9008:08,90,02,A0,07,84,55,AD,02C7
9010:9A,9F,AC,9B,9F,38,E9,03,0443
9018:85,5D,98,E9,00,85,5E,A4,03EA
9020:5D,CC,98,9F,ED,99,9F,B0,0535
9028:0D,A2,00,A0,00,A9,80,8D,0305
9030:B8,9F,2C,B8,9F,60,A0,00,03DA
9038:B1,5D,C9,40,29,07,A8,90,037F
9040:02,69,01,49,FF,65,5D,85,02FB
9048:5F,A9,FF,65,5E,85,60,C4,0473
9050:55,D0,42,B1,63,29,7F,D1,03F4
9058:5F,D0,3A,88,10,F5,C8,B1,046F
9060:5D,A4,55,C8,0A,90,0F,AD,0374
9068:A7,9F,D1,5F,D0,27,C8,AD,04E2
9070:A8,9F,D1,5F,D0,1F,10,09,037F
9078:B1,5F,F0,05,CD,9F,9F,D0,04E0
9080:14,A0,00,B1,5D,4A,8D,B8,0351
9088:9F,C8,B1,5D,AA,C8,B1,5D,04F5
9090:A8,2C,B8,9F,60,A5,5F,A4,0433
9098:60,4C,15,90,20,82,7E,C9,033A
90A0:5C,D0,08,AD,B8,9F,09,40,0381
90A8:8D,B8,9F,20,41,8F,88,C0,041C
90B0:08,90,02,A0,07,84,55,0E,0228
90B8:B8,9F,98,0D,B8,9F,8D,B8,0498
90C0:9F,38,AD,98,9F,E9,03,85,042C
90C8:5D,AD,99,9F,E9,00,85,5E,040E
90D0:AD,B8,9F,C9,28,98,90,02,041F
90D8:69,01,49,FF,65,5D,85,5F,0358
90E0:AA,A9,FF,65,5E,85,60,EC,04E6
90E8:96,9F,ED,97,9F,90,34,A5,04C1
90F0:60,8E,98,9F,8D,99,9F,B1,049B
90F8:63,29,7F,91,5F,88,10,F7,038A
9100:A4,55,C8,2C,B8,9F,10,0C,0360
9108:AD,A7,9F,91,5F,C8,AD,A8,0500
9110:9F,91,5F,B8,50,05,AD,9E,03E7
9118:9F,91,5F,A0,00,AD,B8,9F,0433
9120:91,5D,60,A2,03,4C,98,7E,0355
9128:A0,08,20,DD,7E,A9,00,85,0351
9130:61,20,36,91,C6,61,A9,04,031C
9138:85,62,AD,9A,9F,AC,9B,9F,04B3
9140:38,E9,03,85,5D,98,E9,00,0387
9148:85,5E,A4,5D,CC,98,9F,ED,04D4
9150:99,9F,B0,08,A9,0D,20,E7,03AD
9158:85,4C,E7,85,A0,00,B1,5D,03EB
9160:C9,40,29,07,85,55,AA,90,034D
9168:02,69,01,49,FF,65,5D,85,02FB
9170:5F,A9,FF,65,5E,85,60,B1,0460
9178:5D,C9,40,B0,51,A0,02,24,032D
9180:61,10,06,B1,5D,F0,47,D0,038C
9188:04,B1,5D,D0,41,A0,00,B1,0374
9190:5F,20,E7,85,C8,CA,10,F7,0484
9198:A9,20,20,E7,85,A9,3D,20,035B
91A0:E7,85,A9,20,20,E7,85,A0,0461
91A8:02,B1,5D,AA,88,B1,5D,20,0370
91B0:33,7E,A6,55,A9,20,20,E7,037C
91B8:85,E8,E0,0B,90,F8,C6,62,0508
91C0:D0,0C,A9,04,85,62,A9,0D,0326
91C8:20,E7,85,20,A5,88,A5,5F,03DD
91D0:A4,60,4C,40,91,20,03,90,02D4
91D8:86,61,84,62,10,0E,A2,0C,0299
91E0:2C,A2,0D,2C,A2,02,20,88,0253
91E8:8F,4C,B5,8B,A9,20,2C,B8,03C8
91F0:9F,F0,F1,AD,A6,9F,F0,07,0569
91F8:A9,10,2C,B8,9F,F0,E2,BA,04C8
9200:E0,50,B0,0A,A2,15,20,88,0349
9208:8F,A2,00,4C,98,7E,A9,00,033C
9210:8D,03,02,8D,04,02,20,76,01BB
9218:96,20,5C,7E,C9,28,D0,44,0395
9220:20,56,7E,F0,3A,C9,29,F0,0400
9228:3B,20,AD,8D,EE,03,02,AD,0335
9230:03,02,0A,6D,03,02,AA,A5,01D0
9238:69,9D,03,02,A5,6A,9D,04,02BB
9240:02,A9,00,2C,B9,9F,70,02,02A1
9248:A9,10,AC,A6,9F,F0,02,09,03A5
9250:20,9D,05,02,20,82,7E,C9,02AD
9258:20,F0,C5,C9,29,F0,05,A2,045E
9260:06,20,88,8F,A9,00,8D,AD,0320
9268:9F,20,F1,8C,20,A8,8C,A5,0435
9270:63,48,A5,64,48,AD,A7,9F,03EF
9278:48,AD,A8,9F,48,EE,A9,9F,04BA
9280:AD,A9,9F,D0,03,EE,AA,9F,04FF
9288:AE,AA,9F,8D,A7,9F,8E,A8,0500
9290:9F,EE,BB,9F,A5,61,85,63,04D5
9298:A5,62,85,64,20,82,7E,C9,03D9
92A0:28,D0,5F,20,56,7E,F0,5A,0395
92A8:C9,29,F0,56,20,6D,8F,B0,0404
92B0:51,20,03,90,30,09,A9,20,0206
92B8:2C,B8,9F,D0,3B,F0,03,20,03A1
92C0:9C,90,A0,00,EE,04,02,AD,036D
92C8:04,02,CD,03,02,F0,0F,90,0267
92D0:0D,A9,10,91,5D,98,C8,91,03A5
92D8:5D,C8,91,5D,B0,1A,0A,6D,0354
92E0:04,02,AA,B1,5D,29,CF,1D,02D3
92E8:05,02,91,5D,C8,BD,03,02,027F
92F0:91,5D,C8,BD,04,02,91,5D,0367
92F8:20,76,96,20,82,7E,C9,20,0335
9300:F0,A1,AD,04,02,CD,03,02,0316
9308:F0,05,A2,13,20,88,8F,4C,032D
9310:B5,8B,AD,19,93,AC,1A,93,03F2
9318:2C,29,9D,85,5D,84,5E,A0,0356
9320:00,B1,5D,F0,26,30,22,A0,0316
9328:02,B1,63,29,7F,D1,5D,D0,03BC
9330:30,88,10,F5,A0,03,B1,5D,036E
```

```
9338:8D,AE,9F,B1,63,F0,08,C9,04AF
9340:20,F0,04,C9,3B,D0,1A,18,031A
9348:60,38,60,C8,B1,5D,0A,0A,02E2
9350:0A,0A,8D,AD,9F,C8,B1,5D,03C3
9358:8D,B0,9F,C8,B1,5D,8D,B1,04F0
9360:9F,18,A5,5D,A4,5E,69,04,0328
9368:90,B1,C8,B0,AE,18,A5,63,0487
9370:69,03,85,63,90,02,E6,64,0330
9378:A2,00,20,5C,7E,F0,27,C9,037C
9380:3B,F0,23,C9,23,D0,22,20,034C
9388:56,7E,C9,22,F0,04,C9,27,03A3
9390:D0,0F,20,7C,7E,D0,02,A9,0374
9398:20,85,69,A9,00,85,6A,F0,0396
93A0:03,20,AD,8D,A2,04,4C,3B,028A
93A8:94,C9,28,D0,41,20,56,7E,038A
93B0:20,AD,8D,20,5C,7E,C9,29,0346
93B8:D0,1C,A2,0C,20,56,7E,F0,037E
93C0:7A,C9,3B,F0,76,C9,2C,D0,04A9
93C8:5C,20,7C,7E,29,7F,C9,59,0340
93D0:D0,53,A2,0E,D0,5C,C9,2C,03F4
93D8:D0,4B,20,7C,7E,29,7F,C9,03A6
93E0:58,D0,42,20,7C,7E,C9,29,0376
93E8:D0,3B,A2,10,D0,44,C9,41,03DB
93F0:D0,10,A2,02,A0,01,B1,63,0339
93F8:F0,41,C9,20,F0,34,C9,3B,0442
9400:F0,30,20,AD,8D,A2,06,20,0342
9408:5C,7E,F0,2F,C9,3B,F0,2B,0418
9410:C9,2C,D0,11,20,7C,7E,29,0319
9418:7F,A2,08,C9,58,F0,13,A2,03EF
9420:0A,C9,59,F0,0D,A2,07,20,02F2
9428:88,8F,A9,00,8D,AD,9F,4C,03E5
9430:ED,94,20,56,7E,F0,04,C9,0432
9438:3B,D0,EA,BD,65,9E,2D,B0,0492
9440:9F,85,5D,BD,66,9E,2D,B1,0420
9448:9F,85,5E,2C,B9,9F,50,0F,0365
9450:A2,00,06,5E,26,5D,B0,1F,0258
9458:E8,E0,10,90,F5,B0,0B,A2,04BA
9460:0F,46,5D,66,5E,B0,10,CA,0300
9468:10,F7,A2,07,20,88,8F,A9,0390
9470:00,8D,AD,9F,4C,ED,94,86,042C
9478:5D,E0,07,D0,3B,A0,00,2C,031B
9480:B9,9F,10,06,84,69,84,6A,0349
9488:30,2E,A5,69,ED,AB,9F,08,03AB
9490:38,E9,02,85,69,A5,6A,84,03A4
9498:6A,ED,AC,9F,28,E9,00,D0,0483
94A0:0F,24,69,10,13,A2,09,20,018A
94A8:88,8F,A9,00,85,69,F0,08,03A6
94B0:C9,FF,D0,F1,24,69,10,ED,0513
94B8:18,A5,5D,6D,AD,9F,AA,AD,042A
94C0:AE,9F,5D,45,9E,8D,AE,9F,0467
94C8:A6,5D,BD,77,9E,8D,AD,9F,04AE
94D0:AA,A5,69,8D,AF,9F,A5,6A,04A2
94D8:8D,B0,9F,E0,02,D0,0E,2C,03C8
94E0:B9,9F,30,09,A5,6A,F0,05,0395
94E8:A2,08,20,88,8F,4C,BA,8B,0372
94F0:20,43,95,90,08,A2,11,20,0263
94F8:88,8F,4C,B5,8B,98,18,65,03B8
9500:63,85,63,90,02,E6,64,A5,03CC
9508:5D,0A,65,5D,AA,BD,18,95,033D
9510:48,BD,17,95,48,60,2C,83,0308
9518:96,2C,B2,97,2C,55,97,2C,034F
9520:19,98,2C,57,98,2C,B3,99,0344
9528:2C,3A,99,2C,9F,95,2C,70,02FB
9530:95,2C,E5,98,2C,28,9A,2C,0358
9538:CB,9A,2C,E2,9A,2C,F9,9A,04CC
9540:2C,45,9B,20,7C,7E,A2,00,02C8
9548:86,5D,CA,A0,FF,E8,C8,B1,05AD
9550:63,29,7F,5D,50,9B,F0,F5,0438
9558:29,7F,F0,11,E6,5D,BD,50,03F9
9560:9B,30,03,E8,D0,F8,BD,51,048C
9568:9B,D0,E0,38,60,A5,5D,18,03FD
9570:60,AD,BB,9F,F0,14,68,8D,0460
9578:A8,9F,68,8D,A7,9F,68,85,046F
9580:64,68,85,63,CE,BB,9F,4C,0428
9588:C0,8B,A2,12,20,88,8F,4C,0382
9590:B5,8B,A2,06,2C,A2,0F,2C,02F1
9598:A2,10,20,88,8F,4C,2B,96,02F6
95A0:20,56,7E,F0,F0,20,6D,8F,03F0
95A8:B0,EB,C9,5C,F0,E7,20,12,04C9
95B0:93,90,E2,20,03,90,2C,B8,039C
95B8:9F,30,16,AE,A6,9F,F0,D8,04A0
95C0:A9,20,2C,B8,9F,F0,D1,A0,04AD
95C8:00,B1,5D,09,20,91,5D,D0,02F5
95D0:08,A9,20,8D,B8,9F,20,9C,0371
95D8:90,20,76,96,20,5C,7E,A0,0356
95E0:01,A5,63,91,5D,C8,A5,64,03C8
95E8:91,5D,20,82,7E,F0,3C,C9,0403
95F0:3B,F0,38,C9,28,D0,9E,20,03E2
95F8:56,7E,F0,99,C9,29,F0,2B,046A
9600:C9,5C,F0,16,20,6D,8F,B0,03F7
9608:89,20,03,90,A9,14,2C,B8,02DD
9610:9F,30,07,F0,05,A2,02,20,028F
9618:88,8F,20,76,96,20,82,7E,0363
9620:C9,20,F0,D3,C9,29,F0,03,0491
9628:4C,95,95,A9,00,8D,AD,9F,03F8
9630:20,9D,9A,B0,16,C9,0A,F0,03E0
9638:12,C9,07,F0,0B,C9,05,F0,039B
9640:07,C9,08,D0,EB,4C,B5,8B,041F
9648:A2,1A,2C,A2,14,20,88,8F,02D5
9650:A2,00,4C,98,7E,A0,01,B1,0356
9658:63,8D,A2,9F,C8,B1,63,8D,049A
9660:A3,9F,A5,64,8D,A5,9F,A5,04C1
9668:63,8D,A4,9F,18,69,03,85,033C
9670:63,90,02,E6,64,60,20,41,0300
9678:8F,98,18,65,63,85,63,90,037F
9680:02,E6,64,60,A9,00,85,61,033B
9688:A9,03,20,B7,96,AD,AD,9F,0412
9690:D0,05,A2,04,20,88,8F,20,02D2
9698:F1,8C,20,A8,8C,24,61,70,03C6
96A0:13,A9,03,20,B7,96,AD,AD,0386
96A8:9F,F0,09,20,F1,8C,20,FD,0452
96B0:97,4C,9D,96,4C,C0,8B,85,0432
96B8:62,A9,00,8D,AD,9F,24,61,0369
96C0:30,43,20,56,7E,F0,68,90,034F
```

```
96C8:05,20,36,97,B0,2C,20,AD,029B
96D0:8D,2C,B9,9F,30,07,70,05,02BD
96D8:A2,08,20,88,8F,A5,69,EE,03DD
96E0:AD,9F,AE,AD,9F,9D,AD,9F,052F
96E8:20,82,7E,F0,42,C9,3B,F0,0446
96F0:3E,C9,20,D0,35,C6,62,D0,0424
96F8:C5,60,C9,3B,F0,31,85,5F,042E
9700:06,61,38,66,61,20,7C,7E,0280
9708:F0,20,C5,5F,F0,0B,EE,AD,04CA
9710:9F,AE,AD,9F,9D,AD,9F,D0,0552
9718:DC,06,61,46,61,20,7C,7E,0304
9720:F0,0D,C9,3B,F0,09,C9,20,03E3
9728:F0,98,A2,06,20,88,8F,38,039F
9730:66,61,46,61,38,60,C9,2A,02F9
9738:F0,1A,C9,3C,F0,16,C9,3E,041C
9740:F0,12,C9,24,F0,0E,C9,25,03DB
9748:F0,0A,20,6D,8F,90,05,20,02CB
9750:7F,8F,B0,01,18,60,20,56,02AD
9758:7E,F0,0C,C9,3B,F0,08,20,0396
9760:36,97,90,0B,A2,06,2C,A2,02DE
9768:04,20,88,8F,4C,B5,8B,20,02E7
9770:AD,8D,A5,69,8D,AE,9F,A5,04C7
9778:6A,8D,AF,9F,A9,02,8D,AD,042A
9780:9F,20,F1,8C,20,A8,8C,20,03B0
9788:5C,7E,F0,24,C9,3B,F0,20,0402
9790:20,36,97,B0,16,20,AD,8D,030D
9798:A5,69,8D,AE,9F,A5,6A,8D,0484
97A0:AF,9F,20,F1,8C,20,FD,97,049F
97A8:4C,87,97,A2,06,20,88,8F,0349
97B0:4C,C0,8B,20,56,7E,F0,AF,042A
97B8:C9,3B,F0,AB,20,36,97,B0,043C
97C0:A3,20,AD,8D,A5,6A,8D,AE,0447
97C8:9F,A5,69,8D,AF,9F,A9,02,0433
97D0:8D,AD,9F,20,F1,8C,20,A8,043E
97D8:8C,20,5C,7E,F0,D2,C9,3B,044C
97E0:F0,CE,20,36,97,B0,C4,20,043F
97E8:AD,8D,A5,6A,8D,AE,9F,A5,04C8
97F0:69,8D,AF,9F,20,F1,8C,20,0401
97F8:FD,97,4C,D9,97,AD,A6,9F,0542
9800:F0,17,AD,BB,9F,F0,07,A9,04AE
9808:10,2C,BA,9F,F0,0B,AD,BA,03F7
9810:9F,4A,90,05,A9,0D,4C,E7,0367
9818:85,60,AD,BA,9F,29,BF,8D,0460
9820:BA,9F,20,56,7E,F0,2E,C9,0434
9828:3B,F0,2A,20,36,97,B0,0B,02FD
9830:20,AD,8D,2C,B9,9F,10,0D,02FB
9838:A2,0D,2C,A2,06,20,88,8F,02BA
9840:A2,00,4C,98,7E,A6,69,A4,03B7
9848:6A,86,65,84,66,AD,BA,9F,0445
9850:09,40,8D,BA,9F,4C,B5,8B,03BB
9858:2C,BA,9F,10,0B,0E,BA,9F,0307
9860:4E,BA,9F,A2,02,20,57,89,034B
9868:20,56,7E,F0,43,C9,3B,F0,041B
9870:3F,A5,63,D0,02,C6,64,C6,0409
9878:63,20,CC,88,86,5D,20,ED,03C7
9880:88,D0,05,A2,0A,4C,98,7E,036B
9888:AD,A6,9F,F0,23,A5,5D,A0,04A7
9890:01,A2,02,8E,C0,9F,20,08,02BA
9898:89,AD,AB,9F,20,92,89,AD,0468
98A0:AC,9F,20,92,89,A5,90,D0,048B
98A8:0A,0E,BA,9F,38,6E,BA,9F,0370
98B0:4C,B5,8B,A2,0B,20,88,8F,0370
98B8:A2,02,20,57,89,4C,B5,8B,0330
98C0:A9,01,85,61,A9,00,8D,AD,0373
98C8:9F,20,9D,9A,B0,1B,C9,0A,0394
98D0:F0,1A,C9,06,F0,19,C9,07,03B2
98D8:F0,23,C9,08,F0,2E,C9,09,03D4
98E0:D0,E7,C6,61,D0,E3,4C,B5,0592
98E8:8B,4C,34,9A,4C,29,9A,E6,039A
98F0:61,D0,D6,A2,19,20,88,8F,03F9
98F8:A2,00,4C,98,7E,20,9D,9A,035B
9900:B0,E7,C9,0A,F0,E6,C9,08,0511
9908:F0,BF,D0,F1,20,F1,8C,20,052D
9910:A8,8C,4C,71,95,AA,A9,00,03D9
9918:E0,3E,F0,0A,E0,3D,F0,05,042A
9920:E0,3C,D0,04,2A,2A,2A,60,02CE
9928:68,68,A2,06,2C,A2,04,2C,0276
9930:A2,0D,2C,A2,05,20,88,8F,02B9
9938:4C,B5,8B,20,56,7E,F0,ED,045D
9940:C9,3B,F0,E9,20,36,97,B0,047A
9948:E1,20,AD,8D,2C,B9,9F,30,03EF
9950:DF,20,82,7E,F0,D7,C9,20,04AF
9958:D0,D9,20,56,7E,F0,CE,20,047B
9960:15,99,85,61,20,7C,7E,F0,039E
9968:C4,C9,20,F0,12,20,15,99,037D
9970:24,61,D0,BF,05,61,85,61,0360
9978:20,7C,7E,C9,20,D0,B4,20,03A7
9980:56,7E,20,36,97,B0,A6,A5,03BC
9988:6A,48,A5,69,48,20,AD,8D,0362
9990:38,68,E5,69,85,69,68,E5,0429
9998:6A,2C,B9,9F,30,92,05,69,031E
99A0:F0,06,A9,00,B0,03,38,2A,02B4
99A8:2A,2A,24,61,D0,03,4C,C0,02B8
99B0:98,4C,B5,8B,A9,20,2C,BA,03D3
99B8:9F,D0,5E,0D,BA,9F,8D,BA,047A
99C0:9F,A5,64,48,A5,63,48,20,0360
99C8:CC,88,86,61,20,ED,88,F0,04C0
99D0:4B,18,AD,9C,9F,69,FF,85,0438
99D8:5D,AD,9D,9F,69,FF,85,5E,0491
99E0:AD,98,9F,85,5F,AD,99,9F,04AD
99E8:85,60,20,F1,89,EE,9F,9F,04AB
99F0:F0,2D,AD,9F,9F,8D,9E,9F,04D2
99F8:18,AD,9C,9F,69,FF,85,63,0450
9A00:AD,9D,9F,69,FF,85,64,A0,04DA
9A08:00,B1,63,F0,27,8C,AD,9F,0403
9A10:20,F1,8C,20,A8,8C,4C,31,036E
9A18:8B,A2,16,2C,A2,17,2C,A2,02F6
9A20:18,20,88,8F,A2,00,4C,98,02D5
9A28:7E,A9,00,8D,AD,9F,20,F1,0411
9A30:8C,20,A8,8C,A9,20,2C,BA,038F
9A38:9F,F0,14,4D,BA,9F,8D,BA,0490
9A40:9F,A9,00,8D,9E,9F,68,85,03FF
9A48:63,68,85,64,4C,C0,8B,18,0363
9A50:AD,9C,9F,8D,96,9F,69,FF,0512
```

```
9A58:85,5D,AD,9D,9F,8D,97,9F,048E
9A60:69,FF,85,5E,A9,00,A8,91,042D
9A68:5D,A0,06,20,DD,7E,18,A9,033F
9A70:31,6D,A6,9F,20,E7,85,A9,0418
9A78:2C,20,E7,85,A9,20,20,E7,0388
9A80:85,AD,B6,9F,AE,B7,9F,20,04AB
9A88:CF,7D,A0,07,20,DD,7E,AD,041B
9A90:AB,9F,AE,AC,9F,20,33,7E,0414
9A98:A9,0D,4C,E7,85,20,F1,8C,040B
9AA0:20,A8,8C,20,EA,8F,B0,23,03C0
9AA8:20,55,96,20,82,7E,D0,0D,0308
9AB0:20,7C,7E,C9,2E,F0,10,20,0331
9AB8:76,96,4C,C0,9A,20,7C,7E,03CC
9AC0:20,5C,7E,C9,2E,D0,D6,20,03B7
9AC8:43,95,18,60,A2,01,20,56,0269
9AD0:7E,29,7F,C9,4D,D0,02,A2,03B0
9AD8:11,8A,0D,BA,9F,8D,BA,9F,03E7
9AE0:4C,B5,8B,A2,EE,20,56,7E,0410
9AE8:29,7F,C9,4D,D0,02,A2,EF,0421
9AF0:8A,2D,BA,9F,8D,BA,9F,4C,0442
9AF8:B5,8B,AD,A6,9F,D0,03,20,0425
9B00:05,9B,4C,C0,8B,20,56,7E,032B
9B08:F0,25,C9,3B,F0,21,20,36,0380
9B10:97,B0,22,20,AD,8D,2C,B9,03A8
9B18:9F,10,08,A9,3F,20,E7,85,032B
9B20:20,E7,85,A5,69,A6,6A,20,03CA
9B28:33,7E,20,5C,7E,D0,DB,A9,03FF
9B30:0D,4C,E7,85,60,85,61,20,032B
9B38:7C,7E,F0,F3,C5,61,F0,C5,05B8
9B40:20,E7,85,4C,37,9B,20,A5,036F
9B48:86,4C,C0,8B,FF,FF,FF,FF,0619
9B50:42,59,D4,44,42,59,D4,57,0379
9B58:4F,52,C4,4F,42,CA,4F,55,0364
9B60:D4,46,49,4C,C5,49,C6,44,03C7
9B68:45,C6,5E,44,45,C6,5E,49,035F
9B70:C6,45,4E,C4,4C,49,53,D4,03D9
9B78:4E,4F,4C,49,53,D4,50,52,02FB
9B80:49,4E,D4,50,41,47,C5,00,0308
9B88:93,0D,2A,2A,2A,20,43,3D,01BE
9B90:36,34,20,4D,41,43,52,4F,01FC
9B98:2D,41,53,53,45,4D,42,4C,0234
9BA0:45,52,2F,45,44,49,54,4F,023B
9BA8:52,20,2A,2A,2A,0D,20,20,013D
9BB0:20,20,56,45,52,53,49,4F,0218
9BB8:4E,20,35,2F,38,34,0D,0D,0158
9BC0:00,4F,4B,2E,0D,00,20,42,0137
9BC8:59,54,45,53,20,46,52,45,0242
9BD0:45,2E,0D,00,20,45,52,52,0189
9BD8:4F,52,00,20,49,4E,20,00,0178
9BE0:2D,2D,2D,50,41,47,45,20,01C4
9BE8:00,4C,49,4E,45,20,44,45,01D1
9BF0:4C,45,54,45,44,0D,00,45,01C0
9BF8:4E,44,20,4F,46,20,50,41,01F8
9C00:53,53,20,00,20,45,52,52,01CF
9C08:4F,52,53,2C,20,2A,20,3D,01C7
9C10:20,00,2D,2D,2D,20,4C,41,0154
9C18:42,45,4C,20,46,49,4C,45,0213
9C20:20,2D,2D,2D,0D,0D,00,42,0103
9C28:52,45,41,CB,53,59,4E,54,02F1
9C30:41,D8,49,4C,4C,45,47,41,02C7
9C38:4C,20,51,55,41,4E,54,49,023E
9C40:54,D9,4F,55,54,20,4F,46,02DA
9C48:20,4D,45,4D,4F,52,D9,44,02BD
9C50:45,4C,49,4D,49,54,45,52,025B
9C58:20,45,58,50,45,43,54,45,022E
9C60:C4,4C,49,4E,45,20,54,4F,02AF
9C68:4F,20,4C,4F,4E,C7,44,45,02A8
9C70:56,49,43,45,20,4E,4F,54,0238
9C78:20,50,52,45,53,45,4E,D4,02C1
9C80:49,4C,4C,45,47,41,4C,20,021A
9C88:44,45,56,49,43,C5,46,49,02BF
9C90:4C,45,20,4E,4F,54,20,46,0208
9C98:4F,55,4E,C4,20,46,49,4C,02B1
9CA0:45,20,45,58,49,53,54,D3,02C5
9CA8:4D,49,53,53,49,4E,47,20,023A
9CB0:46,49,4C,45,4E,41,4D,C5,02C1
9CB8:4C,49,53,D4,41,55,54,CF,0375
9CC0:44,45,4C,45,54,C5,52,45,02CA
9CC8:4E,55,CD,46,4F,52,4D,41,02E5
9CD0:D4,46,49,4E,C4,52,45,50,035C
9CD8:4C,41,43,C5,43,4F,50,D9,0350
9CE0:4D,4F,56,C5,44,49,D2,4D,0363
9CE8:45,4D,4F,52,D9,42,52,CB,036B
9CF0:42,41,53,49,C3,44,C3,4C,0335
9CF8:4F,41,C4,53,41,56,C5,41,0344
9D00:50,50,45,4E,C4,55,4E,4E,02E8
9D08:45,D7,43,4F,4C,C4,50,41,034F
9D10:47,C5,41,53,53,45,4D,42,02C7
9D18:4C,C5,4C,41,42,45,CC,53,0344
9D20:59,D3,50,52,49,4E,D4,D0,0409
9D28:00,00,00,EE,30,41,44,43,01E6
9D30:61,41,4E,44,21,43,4D,50,0235
9D38:C1,45,4F,52,41,4C,44,41,02B9
9D40:A1,4F,52,41,01,53,42,43,025C
9D48:E1,00,00,6E,30,53,54,41,0267
9D50:81,00,00,6C,80,41,53,4C,024D
9D58:02,4C,53,52,42,52,4F,4C,0222
9D60:22,52,4F,52,62,00,00,6C,01E3
9D68:00,44,45,43,C2,49,4E,43,0268
9D70:E2,00,01,C8,00,43,50,58,0296
9D78:E0,43,50,59,C0,00,01,DA,0367
9D80:00,4C,44,58,A2,00,01,EC,0277
9D88:00,4C,44,59,A0,00,01,58,01E2
9D90:00,53,54,58,82,00,01,68,01EA
9D98:00,53,54,59,80,00,00,48,01C8
9DA0:00,42,49,54,20,00,00,08,0107
9DA8:08,4A,4D,50,40,00,00,08,0137
9DB0:00,4A,53,52,2C,00,00,01,011C
9DB8:00,42,43,43,90,42,43,53,0230
9DC0:B0,42,45,51,F0,42,4D,49,0350
9DC8:30,42,4E,45,D0,42,50,4C,02B3
9DD0:10,42,56,43,50,42,56,53,0226
9DD8:70,00,00,00,40,42,52,4B,018F
9DE0:00,43,4C,43,18,43,4C,44,01BD
```

```
9DE8:D8,43,4C,49,58,43,4C,56,02ED
9DF0:B8,44,45,58,CA,44,45,59,0345
9DF8:88,49,4E,58,E8,49,4E,59,034F
9E00:C8,4E,4F,50,EA,50,48,41,0378
9E08:48,50,48,50,08,50,4C,41,0215
9E10:68,50,4C,50,28,52,54,49,026B
9E18:40,52,54,53,60,53,45,43,0274
9E20:38,53,45,44,F8,53,45,49,02ED
9E28:78,54,41,58,AA,54,41,59,02FD
9E30:A8,54,59,41,98,54,53,58,032D
9E38:BA,54,58,41,8A,54,58,53,0330
9E40:9A,FF,FF,FF,FF,08,04,14,04B6
9E48:FF,0C,1C,18,00,08,00,10,0157
9E50:00,2C,FF,FF,FF,00,04,14,0341
9E58:14,0C,1C,1C,FF,FF,FF,FF,0454
9E60:FF,FF,FF,FF,FF,00,C0,00,05BB
9E68:80,80,00,49,00,24,00,12,017F
9E70:00,00,08,00,20,00,10,02,003A
9E78:02,02,02,03,03,03,02,01,0012
9E80:01,02,02,03,00,03,06,01,0012
9E88:05,01,01,07,02,08,04,09,0025
9E90:0A,0C,02,01,04,0B,0D,01,0036
9E98:0D,0E,01,01,0E,0E,01,00,003A
9EA0:00,00,00,00,00,20,49,4E,00B7
9EA8:20,00,53,59,4E,54,41,58,0207
9EB0:20,45,52,52,4F,52,00,49,01F3
9EB8:4C,4C,45,47,41,4C,20,51,0222
9EC0:55,41,4E,54,49,54,59,00,022E
9EC8:55,4E,44,45,46,27,44,20,01FD
9ED0:4C,41,42,45,4C,00,52,45,01F7
9ED8:44,45,46,27,44,20,4C,41,01E7
9EE0:42,45,4C,00,4D,49,53,53,020F
9EE8:49,4E,47,20,4F,50,45,52,0234
9EF0:41,4E,44,00,54,59,50,45,0215
9EF8:20,4D,49,53,4D,41,54,43,022E
9F00:48,00,49,4C,4C,45,47,41,01F6
9F08:4C,20,41,44,52,2E,20,4D,01DE
9F10:4F,44,45,00,42,52,41,4E,01FB
9F18:43,48,20,4F,55,54,20,4F,0212
9F20:46,20,52,41,4E,47,45,00,01D3
9F28:49,45,43,2D,45,52,52,4F,0236
9F30:52,00,49,4C,4C,45,47,41,0200
9F38:4C,20,4F,50,2D,43,4F,44,020E
9F40:45,00,55,4E,4B,4E,4F,57,0227
9F48:4E,20,50,53,2E,20,4F,50,01FE
9F50:45,52,41,54,49,4F,4E,00,0212
9F58:49,4C,4C,45,47,41,4C,20,021A
9F60:46,57,44,2E,20,52,45,46,020C
9F68:45,52,45,4E,43,45,00,55,0207
9F70:4E,42,41,4C,41,4E,43,45,0234
9F78:44,20,42,4C,4F,43,4B,4D,021C
9F80:41,52,4B,45,52,00,4F,55,0219
9F88:54,20,4F,46,20,4D,45,4D,0208
9F90:4F,52,59,00,00,50,01,50,019B
9F98:00,7A,00,7A,00,50,00,00,0144
9FA0:00,50,00,00,00,50,FF,00,019F
9FA8:00,00,00,00,00,00,00,00,0000
9FB0:00,00,00,00,00,00,00,00,0000
9FB8:00,00,00,00,80,08,14,00,009C
9FC0:00,00,00,00,00,00,00,00,0000
9FC8:00,00,00,00,00,00,00,00,0000
9FD0:00,0A,00,00,00,00,00,00,000A
9FD8:00,00,00,00,00,00,00,00,0000
9FE0:00,00,00,00,00,00,00,00,0000
9FE8:00,00,00,00,00,00,00,00,0000
9FF0:00,00,00,00,00,00,00,00,0000
9FF8:00,00,00,00,00,00,00,00,0000
```

```
10 IFX=0THENPOKE55,0:POKE56,80:CLR:X=1:LOAD"ASSEMBLER",8,1
20 SYS31232
```

CHIP SPECIAL

COMMODORE 64
PROGRAMMI (1)
programmi listati, L. 5.000
ISBN 88 7081 153 0
Sono stati inclusi nel volume 27 diversi programmi per questo conosciutissimo home computer: giochi, grafica, musica, organizzazione, scienza e hobby sono i temi trattati. Una breve descrizione iniziale illustra scopo e funzionamento di ciascun programma.
A richiesta ogni programma può essere fornito su cassetta o disco.

COMMODORE 64
PROGRAMMI (2)
programmi listati, L. 7.000
ISBN 88 7081 199 9
Giochi, musica, grafica, apprendimento, organizzazione e curiosità varie, sono i temi trattati in questo secondo special per Commodore 64. In totale 26 programmi listati e commentati da una descrizione iniziale che consente all'utilizzatore del computer l'approfondimento delle conoscenze di programmazione. I programmi sono disponibili registrati su cassetta o su dischetto.

COMMODORE 64
PROGRAMMI (3)
programmi listati, L. 7.000
ISBN 88 7081 203 0
Questa pubblicazione sul C 64 presenta una serie di nuovi programmi per questo computer tanto popolare. Contiene 17 programmi di giochi, grafica, musica, organizzazione ed hobby. Potete realizzarli per giocare o per risolvere i problemi.

PEEK·POKE
Commodore 64 · Manuale
H. Rügheimer, C. Spanik
L. 8.000
ISBN 88 7081 228 6
Con questo manuale potrete esplorare ed esaminare il campo sconosciuto della memoria del Commodore. Questo libro deve essere una guida, deve spiegare quali siano le locazioni di utilizzo e come impiegarle pienamente. È presente anche una tabella delle parole-chiave e alcune delucidazioni sui termini tecnici.

CHIP libro club

Mittente

Nome e Cognome

Professione

Via e numero

Città

CAP

Partita IVA

NON AFFRANCARE

Affrancatura a carico del destinatario da addebitarsi sul Conto di Credito n. 3076 presso l'Ufficio Postale di MILANO A. D (autorizz. Dir. Prov. PP. TT. di Milano n. 836942 del 27.11.65)

CHIP

Tecniche Nuove
Via Moscova 46/9
20121 Milano

CHIP libro club

Mittente

Nome e Cognome

Professione

Via e numero

Città

CAP

Partita IVA

NON AFFRANCARE

Affrancatura a carico del destinatario da addebitarsi sul Conto di Credito n. 3076 presso l'Ufficio Postale di MILANO A. D (autorizz. Dir. Prov. PP. TT. di Milano n. 836942 del 27.11.65)

CHIP

Tecniche Nuove
Via Moscova 46/9
20121 Milano

Pagina mancante? (retro coupon)

In questo numero

GIOCHI CONCETTUALI	Esagono
GIOCHI DI CARTE	Casinò-Blackjack
GIOCHI D'AVVENTURA	La terra dei 7 anelli
LINGUISTICA	Le coniugazioni latine
GRAFICA	Istogrammi
PUBBLICITÀ	Scritte scorrevoli
TRUCCHI E CONSIGLI	Caricatore HEX Rat-Tape Screensave
STRUMENTO DI SALVATAGGIO	Dottore dei dischetti
TOOL-SET	CHIP Monitor Macro Assembler/Editor